Agilent Technologies
3458Aマルチメータ

ユーザーズ・ガイド

マニュアル・パーツ番号: 03458-90512
Printed in U.S.A

AGILENT TECHNOLOGIESの保証について

AGILENT製品: 3458Aマルチメータ　　　　保証期間: 1年

1. Agilent Technologiesは、Agilentのハードウェア、アクセサリおよびサプライ品に対して、部品および製造上の不具合について上記の期間に渡って保証します。Agilentが保証期間内にこのような不具合の通知を受けた場合、Agilentは、当社の判断により、不具合があると認められた製品を修理または交換します。交換製品は、新品あるいは新品同等品の場合があります。

2. Agilentは、Agilent ソフトウェアが適切にインストールされ使用されているのにもかかわらず、部品および製造上の不具合によってプログラミング命令の実行に失敗することがないことを上記の期間に渡って保証します。Agilentが保証期間内にこのような不具合の通知を受けた場合、Agilentはこのような不具合のためにプログラミング命令を実行しないソフトウェア・メディアを交換します。

3. Agilentは、Agilent製品の動作が中断されないことや、エラーが皆無であることを保証するものではありません。Agilentが妥当な期間内に製品を保証された状態に修理または交換できない場合、お客様は当該製品をすみやか返却して購入金額の返金を請求できます。

4. Agilent製品には、性能において新品と同等の再生品、あるいは偶発的事故で使用されていた再生品が含まれている場合があります。

5. 保証期間は、納入日または、Agilentにより設置された場合には設置日から始まります。お客様の都合でAgilentによる設置を納入日から30日を超えて遅らせた場合、保証は納入から31日目に始まります。

6. (a)不適切または不完全な保守あるいは校正、(b)Agilentが提供したものでないソフトウェア、インタフェース、パーツまたはサプライ品、(c)当社が認めていない改造または誤使用、(d)製品の公開された環境仕様外での使用、ないしは(e)設置場所の不備または不適切な保全から生じた不具合には保証は適用されません。

7. 該当する地域の法律の範囲内で、ここに定める以外の保証は致しかねます。また、製品の特定用途での市場商品価値や適合性に関する保証は致しかねます。

8. Agilentは、有体財産の損害賠償の場合は、1件につき300,000ドルまたは賠償請求対象となった製品に対して支払われた実際の金額のいずれか多い方に対して責任を負い、人の怪我または死亡に対する損害賠償については、管轄の裁判所により不具合のあるAgilent製品が直接の原因となって生じたと判断された範囲内で責任を負います。

9. 該当する地域の法律の範囲内で、上記の保証条項がお客様の唯一かつ限定された保証条項です。上記の場合を除いて、いかなる場合も、AGILENTとそのサプライヤは、データの喪失または直接損害、特別損害、付随的損害、間接損害(失われた利益またはデータを含む)、その他の損害に対して、契約および不法行為に基づく場合、あるいはその他の場合に関係なく責任を負いかねます。

オーストラリアおよびニュージーランドにおける取引の場合:法律上許されている範囲を除き、上記の保証条項によって、本製品のお客様への販売に適用される法定権利が除外、制限または変更されるものではなく、この保証条項はこの権利に付加されます。

米国政府の権利制限

本ソフトウェアおよび文書は、完全に民間の費用で開発されています。これらは、DFARS 252.227-7013(Oct 1988)、DFARS 252.211-7015(May 1991)あるいはDFARS 252.227-7014(Jun 1995)により定義された"商用コンピュータ・ソフトウェア"、FAR 2.101(a)により定義された"商用アイテム"、またはFAR 52.227-19 (Jun 1987)"(ないしは、同等の機関の規定あるいは契約条項)により定義された"制限付きコンピュータ・ソフトウェア"のいずれかに該当するものとして、納入されラインセス供与されます。該当するFARあるいはDFARS条項、または関連する製品のAgilent標準ソフトウェア契約によりこのようなソフトウェアおよび文書に与えられる権利のみが付与されます。

A

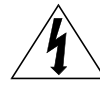
3458Aマルチメータ・ユーザーズ・ガイド
第4版

文書履歴

本マニュアルのすべての版番号および改訂番号と、その作成日を以下にリストします。本マニュアルの最初の版番号は第1版です。版番号は、マニュアルが改訂されるごとに1ずつ増えます。版と版の間に発行される改訂には、マニュアルの現在の版を修正するか、付加情報を追加する代替えページが含まれます。新しい版が作成されると、これには常に前の版の改訂情報のすべてが含まれます。新しい版または新しい改訂版の個々には、訂正されたこの文書履歴ページも含まれます。

第1版 1988月5月
改訂第1版 1992年2月
第2版 1992年10月
第3版 1994年2月
第4版 2000年12月

安全記号

製品に付ける取扱説明書記号。怪我または製品の損傷を防ぐために、ユーザがマニュアルの特定の警告事項または注意事項を参照する必要のあることを示します。

機器を動作させる前にアースに接続する必要のあるフィールド配線端子を示します。過失による感電から保護します。

フレームまたはシャーシのグランド端子。通常は、機器の金属フレームに接続されています。

交流(AC)

直流(DC)

警告、感電の危険

警告　怪我または死亡の原因となる可能性のある手順、方法または条件を示します。

注意　機器の損傷またはデータの永続的な喪失の原因となる可能性のある手順、方法または条件を示します。

警告

本製品の操作、サービスおよび修理時には、次の一般的な安全注意事項を守る必要があります。これらの注意事項、または本マニュアルに別に記載されている特定の警告を守らないと、本製品の設計、製造および意図する使用の安全基準を侵害することになります。Agilent Technologiesは、お客様がこれらの要件を守らなかった場合は責任を負いかねます。

機器は接地してください: 安全クラス1の機器(保護接地端子を備えた機器)の場合、商用電源から製品の入力配線端子または付属電源ケーブルの間に遮断されない安全アース・グランドを設けなければなりません。

爆発性の雰囲気中または可燃性のガスまたは気体のあるところで製品を動作させ**ないでください**。

火災に対して**保護するため**、ライン・ヒューズは電圧および電流の定格とタイプが同じヒューズと交換してください。修理したヒューズを使用したり、ヒューズ・ホルダをショートしたりしないでください。

通電状態の回路からは離れてください。操作者が機器のカバーまたはシールドを取り外さないでください。カバーまたはシールドの取り外しに関する手順は、訓練を受けたサービス員のみが用いるためのものです。特定の条件下では、機器の電源がオフされていても、危険な電圧が存在する可能性があります。危険な感電を避けるために、資格がなければ、カバーまたはシールドに関する手順を実行しないでください。

損傷した機器は動作させないでください: 本製品に内蔵された安全保護機能が、物理的な損傷、過剰な湿気または他の理由で損なわれている可能性がある場合は、常に電源を切り離して、訓練を受けたサービス員によって安全な動作が確認されるまで製品を使用しないでください。必要であれば、製品をAgilentにサービスまたは修理のために返送して、安全機能が維持されていることを確認してください。

一人でサービスまたは調整を行わないでください: 応急処置と蘇生ができる別の人がいなければ、社内でサービスまたは調整を行わないでください。

部品を交換したり、機器を改造したりしないでください: さらなる危険を招くため、部品を交換したり、製品に対して許可されていない改造を加えたりしないでください。必要であれば、製品をAgilentにサービスまたは修理のために返送して、安全機能が維持されていることを確認してください。

高電圧の測定は常に危険です: いずれかの入力端子に42V(DCまたはピーク)を超える入力が接続されている場合には、すべてのマルチメータ入力端子(前面パネルと裏面パネルの両方)を危険と見なす必要があります。

150VAを超える電力を供給できる危険な電圧または電源を**固定して配線する場合**は、ラベルを付ける、ヒューズを入れる、または他の方法で偶然の接触や機器の故障から保護する必要があります。

測定端子は、使用していない場合には通電状態のままに**しないでください**。

前面/裏面パネル・スイッチを使用して、マルチメータの前面端子と裏面端子間で危険な信号を多重化**しないください**。

	DECLARATION OF CONFORMITY According to ISO/IEC Guide 22 and CEN/CENELEC EN 45014	

Manufacturer's Name: Agilent Technologies, Incorporated

Manufacturer's Address: 815 14th ST. S.W.
Loveland, CO 80537
USA

Declares, that the product

Product Name: Multimeter
Model Number: 3458A
Product Options: *This declaration covers all options of the above product(s).*

Conforms with the following European Directives:

The product herewith complies with the requirements of the Low Voltage Directive 73/23/EEC and the EMC Directive 89/336/EEC (including 93/68/EEC) and carries the CE Marking accordingly

Conforms with the following product standards:

EMC	Standard	Limit
	IEC 61326-1:1997+A1:1998 / EN 61326-1:1997+A1:1998	
	CISPR 11:1990 / EN 55011:1991	*Group 1 Class A*
	IEC 61000-4-2:1995+A1:1998 / EN 61000-4-2:1995	*4kV CD, 8kV AD*
	IEC 61000-4-3:1995 / EN 61000-4-3:1995	*3 V/m, 80-1000 MHz*
	IEC 61000-4-4:1995 / EN 61000-4-4:1995	*0.5kV signal lines, 1kV power lines*
	IEC 61000-4-5:1995 / EN 61000-4-5:1995	*0.5 kV line-line, 1 kV line-ground*
	IEC 61000-4-6:1996 / EN 61000-4-6:1996	*3V, 0.15-80 MHz I cycle, 100%*
	IEC 61000-4-11:1994 / EN 61000-4-11:1994	*Dips: 30% 10ms; 60% 100ms* *Interrupt > 95%@5000ms*
	Canada: ICES-001:1998	
	Australia/New Zealand: AS/NZS 2064.1	

The product was tested in a typical configuration with Agilent Technologies test systems.

Safety

IEC 61010-1:1990+A1:1992+A2:1995 / EN 61010-1:1993+A2:1995
Canada: CSA C22.2 No. 1010.1:1992
UL 3111-1: 1994

8 March 2001
Date

Ray Corson
Product Regulation Program Manager

For further information, please contact your local Agilent Technologies sales office, agent or distributor.
Authorized EU-representative: Agilent Technologies Deutschland GmbH, Herrenberger Straβe 130, D 71034 Böblingen, Germany

Revision: B.01 Issue Date: March 2001

はじめに

本書には、Agilent 3458Aマルチメータの設置、操作およびプログラミング、設定に関する情報が含まれています。本書は、以下の章から構成されています。

第1章　設置と保守

本章では、受入れ検査、設置および保守について説明します。別売りのオプションおよびアクセサリの一覧も掲載されています。

第2章　入門

本章では、本器の基本的な操作について説明します。フロントパネルの使用方法、リモートからの本器へのコマンドの送信方法、リモートからのデータの取得方法を紹介します。

第3章　測定のための構成

本章では、あらゆる種類の測定を実現できるように本器を構成するための方法について説明します（ただし、ディジタイズは除きます。ディジタイズについては、第5章で説明します)。さらに、サブプログラム/ステート・メモリ、入力バッファおよびステータス・レジスタの使用方法についても説明します。

第4章　測定の実行

本章では、測定のトリガ方法、読み取り値フォーマット、読み取り値メモリの使用方法、GPIBバスによる測定値の転送方法について説明します。さらに、読み取り速度を上げたり、本マルチメータのEXTOUT信号や演算機能を使用するための方法についても説明します。

第5章　ディジタイズ

ディジタイズとは、連続するアナログ信号を一連の離散サンプル(読り取り値)に変換するプロセスのことを言います。本章では、信号をディジタイズするための各種方法、サンプリング・レートの重要性、レベル・トリガ機能の使用方法について説明します。

第6章　コマンド・リファレンス

本章では、本マルチメータの言語(HPML)について説明する他、HPML言語のコマンドについて1つ1つ詳細に説明します。コマンドは、アルファベット順に示されています。

第7章　BASICプログラミング言語

本章では、Agilent 3458Aに内蔵されているBASIC言語オペレーティング・システムによってサポートされているBASICコマンドについて説明します。この機能を用いれば、本器の動作をカスタマイズするための簡単なBASICサブプログラムを記述し、ダウンロードすることによって、ユーザ独自の要件のほとんどに簡単に対応できます。

付録

付録には、本器の仕様、本器が認識するGPIBコマンドに関する情報、前面/裏面端子スイッチのロック方法に関する情報が含まれている他、ディジタイズや本器の読み取り速度およびスループットの向上方法に関するProduct Noteが含まれています。

目次

第4章　測定の実行

第5章　ディジタイジング

第6章　コマンド・リファレンス

第7章　3458A用のBASIC言語

付録A　仕様

付録B　GPIBコマンド

付録C　前面/裏面端子スイッチとガード端子スイッチのロック

付録D　スループットと測定速度の最適化

付録E　3458Aを使用した高分解能ディジタイズ

第1章 設置と保守

第1章　設置と保守

概要

この章では、受入れ検査、設置および保守について説明します。別売りのオプションおよびアクセサリの一覧も掲載されています。本章をお読みになってから、電気的な接続を行ってください。

受入れ検査

警告　**次のような徴候が見られる場合、または予想される場合には、本器を稼動させないでください。**

1. **目に見える損傷**
2. **輸送時の強度な圧力**
3. **悪条件のもとでの長期間にわたる保管**
4. **意図する測定または機能の実行の失敗**

訓練を受けた修理技術者によって安全に動作することが確認されるまでは、本器を使用しないでください。

本マルチメータは、工場で入念な検査を受けた上で出荷されているため、受取り時には損傷がなく、正常に動作するはずです。輸送用カートンまたは緩衝材に損傷がある場合には、梱包の中身をチェックし、マルチメータの検査を行うまで、保管しておいてください。開梱時には、本ユーザーズ・ガイドに加えて、以下の品目が含まれていることを確認してください。

- クイック・リファレンス・ガイド(1冊)
- フロント・パネル操作ユーザーズ・ガイド(1冊)
- 校正マニュアル(1冊)
- アセンブリレベル修理マニュアル(1冊)
- 電源コード(1本)
- 交換用電源ヒューズ: 500mA T(220/240動作の場合1個)、1.5A NTD(100/120動作の場合1個)
- キーボード・オーバレイ(2個)
- スイッチ・ロックアウト・キャップ(2個)
- テスト・リード・キット(1個)

マルチメータが損傷している場合、または中身が全部揃っていない場合には、すぐに最寄のAgilent Technologies営業所までご連絡ください。

オプションおよびアクセサリ

本マルチメータ用の別売りオプションを表1に、別売りアクセサリを表2にそれぞれ示します。

表1. 別売りオプション

内容	オプション番号	後付け用部品番号
拡張読み取り値メモリ(合計148Kバイトまで拡張)	001	03458-87901
高安定度基準(4ppm/年)	002	03458-80002
フロント・ハンドル・キット	907	5063-9226
ラック・フランジ・キット	908	5063-9212
ラック・フランジ・キット(ハンドル付き)	909	5063-9219
引き取りハードウェア・サポートの2年間延長	W30	

表2. 別売りアクセサリ

内容	モデル/部品番号
ユーザーズ・ガイド、クイック・リファレンス・ガイド、校正マニュアル、アセンブリレベル修理マニュアル、フロント・パネル操作ユーザーズ・ガイドの追加	03458-90101
クイック・リファレンス・ガイドの追加	03458-90008
校正マニュアルの追加	03458-90017
アセンブリレベル修理マニュアルの追加	03458-90011
3458Aフロント・パネル操作ユーザーズ・ガイドの追加	03458-90007
ユーザ定義キー・オーバレイ	03458-84313
スイッチ・ロックアウト・キャップ(1個)	03458-44113
GPIBケーブル(1m)	10833A
GPIBケーブル(2m)	10833B
GPIBケーブル(4m)	10833C
GPIBケーブル(0.5m)	10833D
試験用リード線セット	34132B
低温試験用リード線対、スペード・ラグ間、0.9m	11053A
低温試験用リード線対、スペード・ラグ-バナナ間、0.9m	11174A
低温試験用リード線対、バナナ-バナナ間、0.9m	11058A
40kV AC/DC高電圧プローブ	34136A
30A電流シャント	34330A
ケルビン・プローブ・セット(4線、各1m)	11059A
ケルビン・クリップ・セット(各2個)	11062A
サーミスタ温度プローブ5kΩ	E2308A
5kΩサーミスタ	40653B

マルチメータの設置

このセクションでは、マルチメータのグランド条件と電源条件について説明した後で、マルチメータの設置方法について説明します。(スイッチ・ロックアウト・キャップの取付け方法については、付録Cを参照してください)。図1は、マルチメータの裏面パネルを示したものです。裏面パネル上のほとんどのコネクタとスイッチをこのセクションで参照できます。

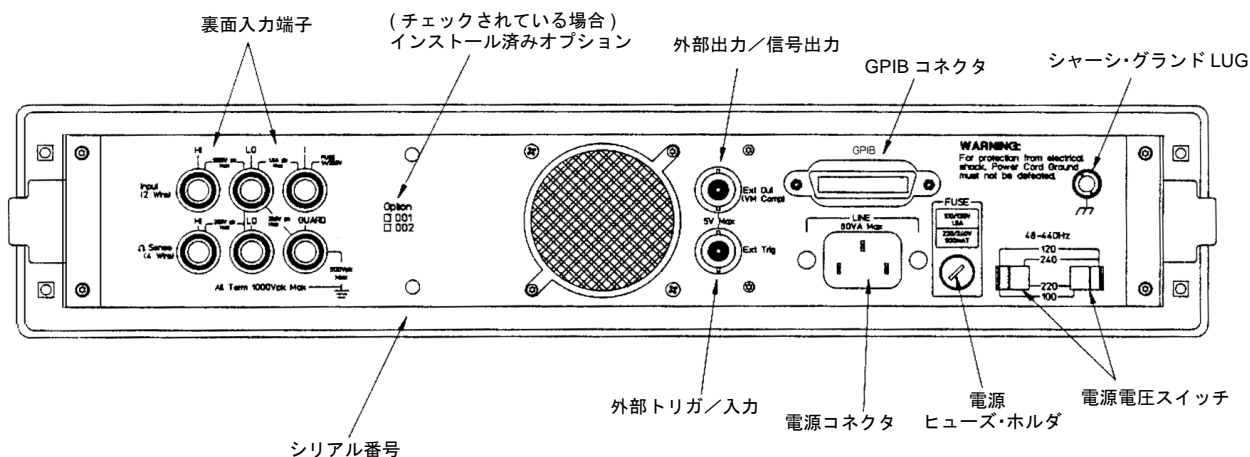



図1. 裏面パネル

グランド条件

本器には、3極AC電源ケーブルが付属しています(図3を参照)。この電源ケーブルは、必ず、グランド・コネクタが電気アース端子(安全用アース端子)に接続された認定3極電気コンセントに接続してください。本器の電源ソケットおよび付属の電源ケーブルは、国際電気技術標準機関(IEC)の安全規格に適合しています。

警告　**感電事故を防ぐため、電源コードのアース端子は絶対に壊さないでください。**

電源要件

本器は、48～440 Hzで、100 VAC、120 VAC、220 VACまたは240 VAC(すべて実効値)を供給する単相電源で動作します。電源電圧は、±10%の範囲で変動可能ですが、250 VACの実効値を超えてはなりません。最大消費電力は80 VA(ボルトアンペア)。表3には、公称電源電圧値と対応する制限値が示されています。

注意　**マルチメータが損傷する可能性があります。本器を AC 電源に接続する前に、本器の電源選択スイッチがAC電源電圧と一致するように設定されていること、適切な電源ヒューズが取り付けられていることを確認してください。詳細については、以下の各セクションを参照してください。**

表3. 電源電圧制限値

公称値(実効値)	許容制限値(実効値)
100 VAC	90 VAC～110 VAC
120 VAC	108 VAC～132 VAC
220 VAC	198 VAC～242 VAC
240 VAC	216 VAC～250 VAC

電源電圧スイッチの設定

電源電圧選択スイッチは、出荷国に応じて事前に設定されています。この設定を変更する場合には、以下の手順に従ってください。

1. AC電源電圧選択スイッチの位置を変更する前に、マルチメータの電源コードを抜きます。
2. 小型のマイナス・ドライバを使って、該当する位置にスイッチを動かします(図2を参照)。
3. 正しい電源ヒューズを取り付けます(次のセクションを参照)。

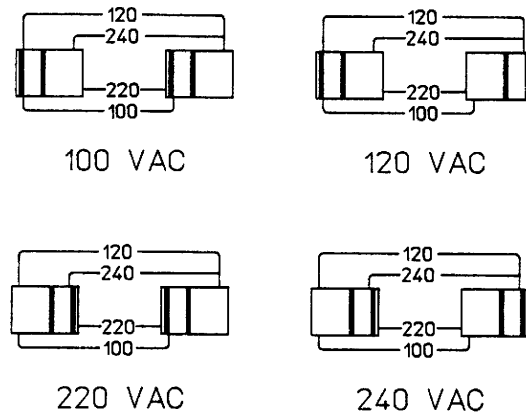



図2. AC電源電圧スイッチの位置

電源ヒューズの取付け

電源ヒューズは、電源電圧の選択と一致していなければなりません。100 VACまたは120 VACで動作する場合は、1.5Aヒューズを取り付けます。220 VACまたは240 VACで動作する場合は、500 mA Tヒューズを取り付けます。

電源ヒューズのホルダは、本器の裏面パネルの右側にあります(図1を参照)。ヒューズを取り付ける場合、本器の電源コードが取り外されていることを確認します。ヒューズの一端をヒューズ・キャップに差し込みます。ヒューズ/キャップ・アセンブリをヒューズ・ホルダに差し込みます。小型のマイナス・ドライバを使って、ヒューズ・キャップを押さえながら、時計回りに回します。

電源コード

図3には、各種マルチメータ電源コードとそれらのAgilent部品番号が示されています。誤った電源コードが入っていた場合には、Agilent営業所に連絡して、交換してください。

電源コード

国名	部品番号	オプション	電圧
オーストラリア	8120-1369	901	250V 6A
デンマーク	1820-2956	912	259V 6A
欧州	1820-1689	902	250V 6A
英国	1820-1351	900	250V 6A
スイス	1820-2104	906	250V 6A
米国	1820-1378	903	120 10A
米国	1820-0698	904	240V 10A

Agilentが提供する電源コードの極性は、機器の電源入力ソケットと一致しています。

注記 : コネクタ側から見たプラグです。成型プラグの形は同一国内でも異なる場合があります。

*CSAが認証しているのは、これらの電源コードだけです。

図3. 電源コード

GPIBケーブルの接続

GPIB[1] ケーブルをマルチメータの裏面パネルにある24ピンGPIBコネクタに接続します。ケーブル・コネクタの2つのネジを手でしっかりと閉めます。図4は、マルチメータとコントローラの間の典型的なGPIB接続を示したものです。

1. GPIBは、IEEE標準488-1978およびANSI MC 1.1を実装したものです。

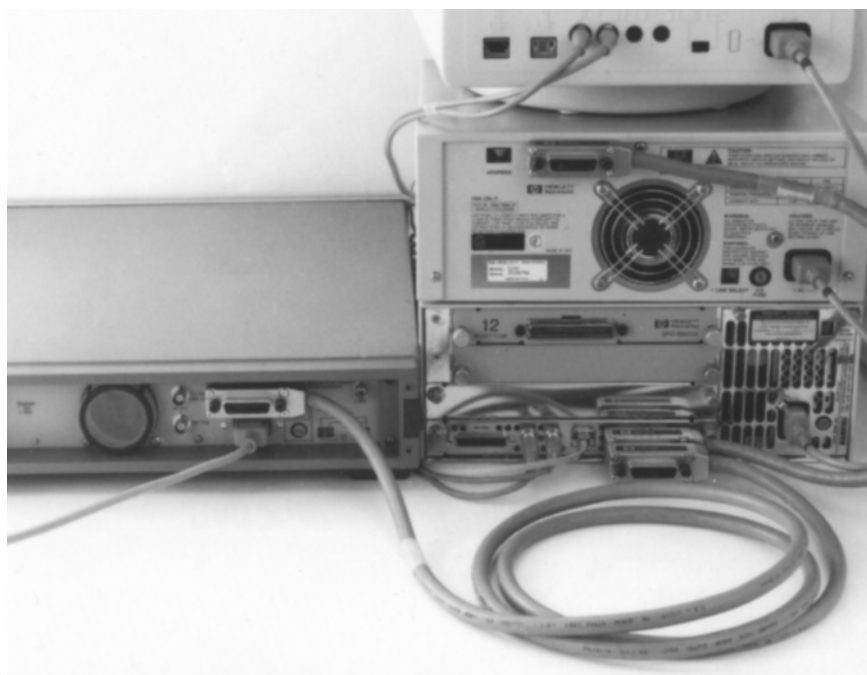



図4. 典型的なGPIB接続

同じGPIBバス上に、全部で15台のデバイスを同時に接続できます。ケーブルの端にはそれぞれ、単一のオス/メス型コネクタが付いているので、数本のケーブルを組み合わせて用いることができます。GPIBケーブルの長さは、全部で20 m、または1台のデバイス当たり2 m(どちらか短い方)を超えてはなりません。

GPIBアドレス

ADDRESSコマンドを使って、マルチメータのGPIBアドレスを変更することができます。GPIBアドレスの変更手順については、第2章の「GPIBアドレスの変更」を参照してください。本器のGPIBアドレスは、工場出荷時には10進法の22に設定されています。対応するASCIIコードは、リスン・アドレス6とトーク・アドレスVです。

注記

本書に示されている例は、BASIC言語を使用するHewlett-Packardシリーズ200/300コンピュータを対象としています。これらの例では、GPIBインタフェース・セレクト・コードが7、機器アドレスが22、GPIBアドレスはそれらを結合して722であると仮定されています。

マルチメータの取付け

本マルチメータには、4本の脚が付属しているので、ベンチ用機器として使用することができます。傾斜スタンドも2本付属しているので、マルチメータの前面を持ち上げることができます。本器は、別売りのラック・マウント・キット(表1を参照)を用いることによって、19インチの標準ラックに取り付けることができます。

設置検査

次のプログラムは、マルチメータが動作しており、GPIBバスを制御するコントローラと通信できることを検証します。

```
10 PRINTER IS 1
20 OUTPUT 722;"ID?"
30 ENTER 722; IDENT$
40 PRINT IDENT$
50 END
```

本マルチメータが正しく設置された場合には、HP 3458Aというメッセージが指定されたシステム・プリンタ上に出力されます。メッセージがプリントされない場合には、マルチメータに電源が供給されているか確かめてください。また、GPIB接続、インタフェース・アドレスの設定、マルチメータのアドレスもチェックしてください。

保守

このセクションでは、マルチメータのヒューズの交換方法と修理サービスを受けるための方法について説明します。

電源ヒューズの交換

電源ヒューズのホルダは、マルチメータの裏面パネルの右側にあります。ヒューズを交換する前に、本器の電源を切断します。ヒューズを交換するには、小型のマイナス・ドライバを使って、ヒューズ・キャップを押さえながら、反時計回りに回します。ヒューズ・キャップを取り外し、ヒューズを適切なタイプ(表4を参照)のものと交換にします。(灰色の電源ヒューズ・キャップのAgilent部品番号は2110-0565です)。ヒューズ・キャップを取り付け直し、電源を入れます。

表4. 交換用電源ヒューズとキャップ

電源電圧	電源ヒューズ
100または120 VAC(公称値)	1.5A NTD、Agilent部品番号2110-0043
220または240 VAC(公称値)	500mA T SB、Agilent部品番号2110-0202

電流ヒューズの交換

前面と裏面にある電流端子(ラベルI)の中にはそれぞれ、電流ヒューズがあります。電流ヒューズを取り出すには、電流端子の留め金ノブのネジを止まるまでゆるめます(反時計回りに回します)。端子を押さえながら、時計回りに回します。これで、端子/ヒューズ・アセンブリ全体を取り外すことができます(図5を参照)。必要であれば、ヒューズを1A 250V NTDヒューズ(Agilent部品番号2110-0001)と交換します。(注意: 絶対に遅延ヒューズを電流ヒューズとして使用しないでください。マルチメータに損傷が及ぶ恐れがあります)。端子/ヒューズ・アセンブリを交換するには、アセンブリがしっかり固定されるまで、押しながら反時計回りに回します。

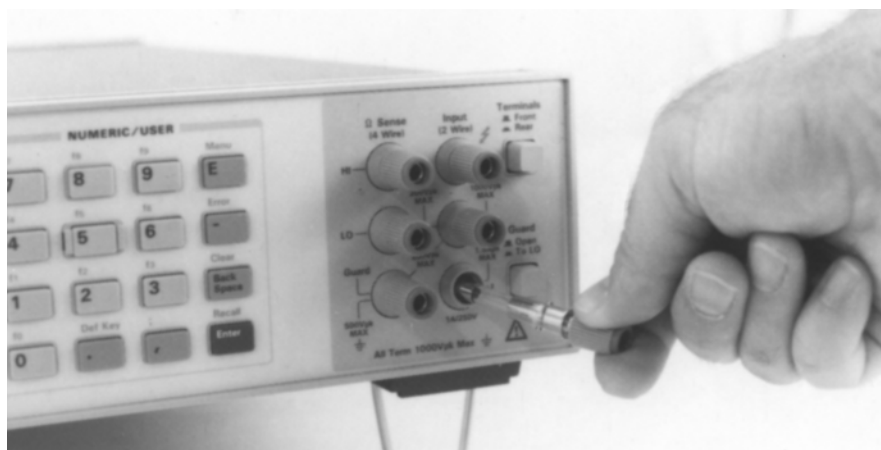



図5. 電流端子/ヒューズ・アセンブリ

修理サービス

本器は、保証期間中であるか否かに関わらず、Agilent Technologiesサービス・センタで修理を受けることができます。本器を送る前に、最寄のAgilent営業所にご連絡になり、輸送方法についてお問い合わせください

シリアル番号

Agilentの機器は、0000A00000という形式の2つの部分から成る10文字のシリアル番号によって識別されます。最初の4桁は、同一製品すべてに関して同じで、その製品に対して変更が加えられた場合にだけ変わります。英字は原産国を示します。Aは、その製品が米国で製造されたことを示します。最後の5桁は、各機器に固有です。本マルチメータのシリアル番号は、マルチメータの裏面にある各種端子の右側にあります。

輸送方法について

本マルチメータを輸送しなければならない場合には、輸送時の損傷を防ぐために、本器が保護用パッケージに入れられていること（納品時の輸送用カートンおよび緩衝用材料を使用していること）を確認してください。輸送時の損傷は、保証の対象範囲外です。荷物には、所有者を明確にするとともに、必要なサービスまたは修理を示すタグを付けてください。本器のモデル番号とシリアル番号も記載してください。荷物には保険を掛けることをお勧めします。

第2章　入門

第2章　入門

概要

この章は、初心者のユーザを対象としています。前面パネルの操作方法、リモートからマルチメータへコマンドを送信する方法、リモートからデータを取り出す方法を説明します。最初に、前面パネルの操作について説明し、電源投入時の状態、表示インジケータ、パラメータを選択/入力するための各種方法、簡単なDC電圧の測定方法など、重要な項目について説明します。したがって、主にリモートからの本器の使用を考えている場合でも、この章全体をお読みください。

電源を入れる前に

- マルチメータの裏面パネルにある電源電圧選択スイッチがその地域の電源電圧と一致するように設定されていることを確認します。
- 適切な電源ヒューズが取り付けられていることを確認します。

取付け条件または電源条件について疑問がある場合には、第1章を参照してください。

電源の投入

本器の電源を入れるには、前面パネルの**Power**スイッチを押します。本器の電源が入らないようであれば、本器が電源に接続されているか確認してください。電源に問題がなければ、電源コードを取り外して、電源ヒューズと電源電圧の選択スイッチの設定を確認してください。

電源投入時セルフテスト

電源を入れると、本器は限定された電源投入時セルフテストを実行します。このテストでは、本器が動作していることは確認されますが、正確な測定が行なえるかどうかは必ずしも検証されません。

電源投入時の状態

電源投入時セルフテストが完了すると、本器はビープ音を1回鳴らし、自動的にトリガをかけ、レンジを自動的に選択し、DC電圧測定を実行します。本器はまた、多くのコマンドの既定の電源投入時の値(表5を参照)への設定を完了します。これは、電源投入時の状態と呼ばれます。

表5. 電源投入時の状態

コマンド	説明
ACBAND 20, 2E6	AC帯域幅20Hz～2MHz
AZERO ON	オートゼロをオンにする
DCV AUTO	DC電圧、オートレンジ
DEFEAT OFF	無効機能をオフにする
DELAY -1	デフォルト遅延
DISP ON	表示をオンにする
EMASK 32767	全エラー条件をオンにする
END OFF	GPIB EOI機能をオフにする
EXTOUT ICOMP, NEG	入力完了EXTOUT信号、負パルス
FIXEDZ OFF	固定入力抵抗をオフにする
FSOURCE ACV	周波数／周期信号源はAC電圧である
INBUF OFF	入力バッファをオフにする
LEVEL 0, AC	0%でのレベル・トリガ、AC結合
LFILTER OFF	レベル・フィルタをオフにする
LFREQ 50 or 60	測定電源周波数を50または60Hzに丸める
LOCK OFF	キーボードをオンにする
MATH OFF	リアルタイム演算をオフにする
MEM OFF	読み取り値メモリをオフにする(直近のメモリ動作 = FIFO)
MFORMAT SREAL	単精度実数読み取り値メモリ・フォーマット
MMATH OFF	後処理演算をオフにする
NDIG 7	7.5桁を表示する
NPLC 10	10電源周波数の積分時間
NRDGS 1, AUTO	トリガ当たりの測定回数1回、オート・サンプリング・イベント
OCOMP OFF	オフセット補正抵抗をオフにする
OFORMAT ASCII	ASCII出力フォーマット
QFORMAT NORM	ノーマル・クエリ・フォーマット
RATIO OFF	比測定をオフにする
RQS 0 (or 8)	0はステータス・レジスタ条件をオフにする(電源切断時に、パワーオンSRQがオンになっている場合、値 = 8)
SETACV ANA	アナログAC電圧モード
SLOPE POS	レベル・トリガの立上がりスロープ
SSRC LEVEL, AUTO	レベル同期信号源イベント、自動同期AC電圧
SWEEP lOOE-9,1024	サンプリング間隔100 ns、1024サンプル
TARM AUTO	オート・トリガ・アーム・イベント
TBUFF OFF	外部トリガのバッファリングをオフにする
TIMER 1	タイマ間隔1秒
TRIG AUTO	オート・トリガ・イベント

以下を除くすべての演算レジスタが0に設定される:
DEGREE = 20 REF=l SCALE = 1 RES=50 PERC = 1

ディスプレイ

電源投入時の状態では、DC電圧が新たに測定される度に、ディスプレイが継続的に更新されます。ディスプレイの下部に沿って、一連のインジケータがあります。これらのインジケータは、さまざまな状態を警告します。例えば、**SMPL**インジケータは、マルチメータが読み取りを完了する度に点灯します。表6には、各ディスプレイ・インジケータの意味が示されています。

表6. ディスプレイ・インジケータ

ディスプレイ・インジケータ	説明
SMPL	読み取りが完了する度に点灯する
REM	マルチメータはGPIBリモート・モードにある
SRQ	マルチメータはGPIBサービス・リクエストを生成した
TALK	マルチメータはGPIBでトークするようにアドレス指定されている
LSTN	マルチメータはGPIBでリスンするようにアドレス指定されている
AZERO OFF	オートゼロはオフである
MRNG	オートレンジはオフである(マルチメータは固定レンジを使用している)
MATH	1回または2回のリアルタイム/後処理演算がオンになっている
ERR	エラーが検出された
SHIFT	シフトキーが押されている
MORE INFO	現在の構成に関する詳細情報が入手可能である (情報を表示するには、右向きの矢印キーを使用)

注記　**ERR**インジケータがこの時点で点灯している場合には、電源投入時セルフテスト中またはその後に、エラーが検出されました。エラーの確認方法については、本章の「エラー・レジスタの読み取り」で後述します。

前面パネルからの操作

このセクションでは、簡単なDC電圧の測定方法、各種前面パネル・キーの使用方法を紹介すると同時に、前面パネルの操作に欠かせないマルチメータの諸機能について説明します。図6は、マルチメータの前面パネルの機能を示したものです。

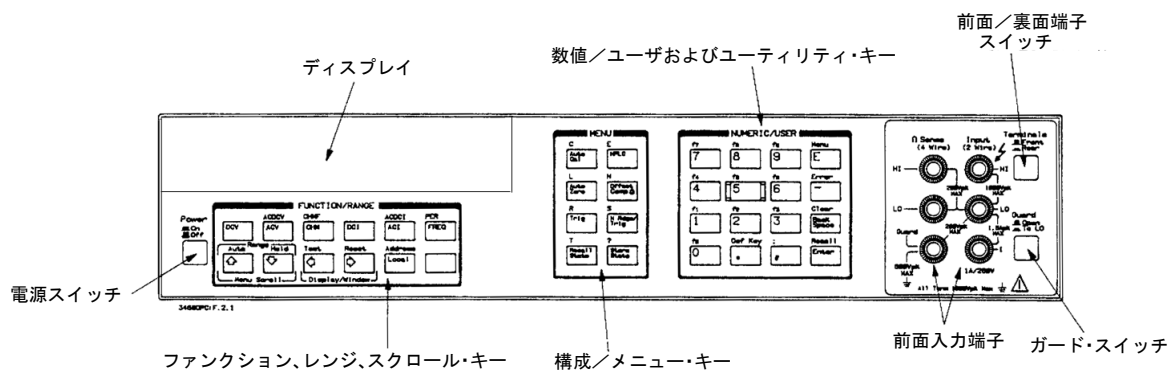



図6. 前面パネル

測定の実行

電源投入時の状態では、DC電圧測定が選択されており、マルチメータは自動的にトリガをかけ、そのレンジを選択します。電源投入時の状態では、図7に示されているようにDC電圧を入力端子に接続するだけで、DC電圧測定を実行できます。図7に示されている接続は、AC電圧、2端子抵抗、AC+DC電圧、ディジタイズ、電圧入力源からの周波数/周期測定にも適用できます。マルチメータの最大入力電圧と最大入力電流に関する注意事項については、第3章を参照してください。

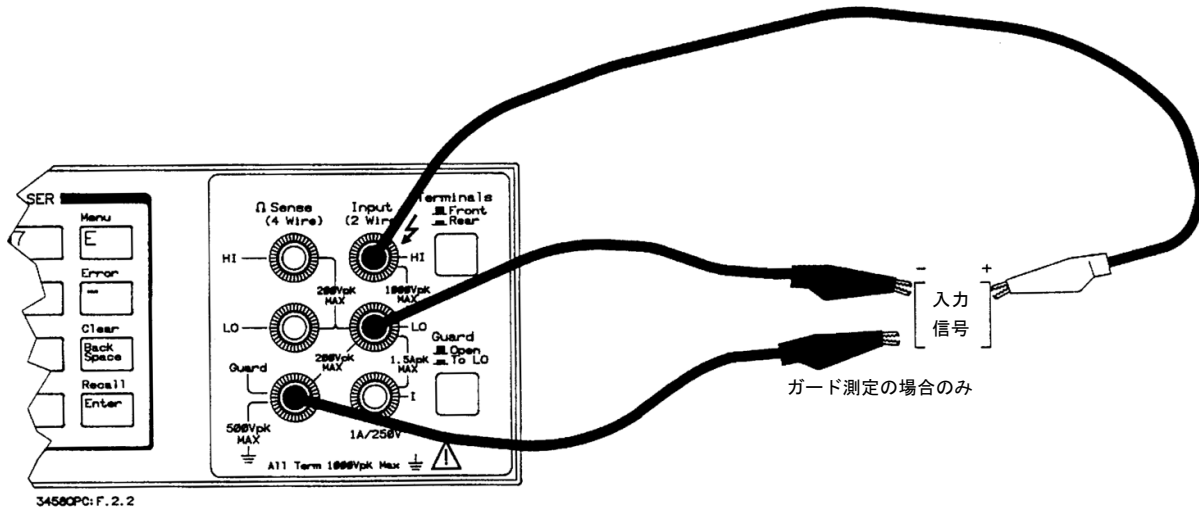



図7. 標準的な2端子(およびガード端子)測定

測定ファンクションの変更

ディスプレイの真下にあるキーの並び(**ファンクション・キー**)で、マルチメータの標準的な測定ファンクションを選択します。表7は、各種**ファンクション・キー**と各ファンクション・キーによって選択される測定ファンクションを示したものです。

表7. ファンクション・キー

キー	説明
DCV	DC 電圧測定
ACV	AC 電圧測定
OHM	2 端子抵抗測定
DCI	DC 電流測定
ACI	AC 電流測定
FREQ	周波数測定
[シフト] ACDCV ACV	AC+DC 電圧測定
[シフト] OHMF OHM	4 端子抵抗測定
[シフト] ACDCI ACI	AC+DC 電流測定
[シフト] PER FREQ	周期測定

本マルチメータは、**ファンクション**・キーによって選択されたファンクションに加えて、直接サンプリングしたディジタイズまたはサブ・サンプリングしたディジタイズ、比測定、同期またはランダム測定法を用いたACまたはAC+DC電圧測定を実行できます。これらの機能は、アルファベット・メニュー・キー (これらのキーについては、このセクションの「メニュー・キーの使用方法」の中で後述)を使って該当するコマンドを呼び出すことによって、前面パネルから選択できます。測定機能または測定方法の詳細については、第1章を参照してください。

オートレンジと手動レンジ切替え

電源投入時の状態では、本マルチメータは適切な測定範囲を自動的に選択します。これはオートレンジと呼ばれます。たいていの場合は、オートレンジを使用し続けたいでしょうが、この他にもホールドと手動レンジ切替えの2種類の選択肢があります。

ホールド

この選択項目を選択すると、オートレンジを停止できます。これを行なうには、オートレンジでレンジが選択されるようにして、以下を押します。

注記

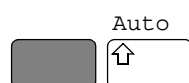
青色のシフトキーを押すと、ディスプレイの**SHIFT**インジケータが点灯します。シフト+キーの機能は、各キーの上に青色で印字されています。

ディスプレイの**MRNG**(手動レンジ)インジケータがオンになっていることに注意してください。このインジケータは、オートレンジを使用していない場合には必ずオンになっています。

手動レンジ切替え

この2番目の選択項目を選択した場合には、レンジを選択できます。マルチメータが測定モードにある(すなわち、マルチメータが測定を実行・表示しているか、ディスプレイにOVLDと表示されている)場合には、上または下矢印キーを押してレンジを変更することが可能です。レンジを増分するには、以下を押します。

上矢印キーを繰り返し押すことによって、最大レンジまで増分できます。最大レンジに達した場合には、上矢印キーを押してもレンジは変わりません。レンジを減分するには、以下を押します。

下矢印キーを繰り返し押すことによって、最小レンジまで減分できます。最小レンジに達した場合には、下矢印キーを押してもレンジは変わりません。オートレンジに戻るには、以下を押します。

セルフテスト

マルチメータの電源を入れると、限定された電源投入時セルフテストが自動的に実行されますが、測定を開始する前に、マルチメータが完全に動作可能状態にあるという確信が欲しい場合もあります。これがセルフテストの役割です。セルフテストでは、マルチメータの動作および確度をチェックする一連のテストが実行されます。

注記

セルフテストを実行する前に、必ず、すべての入力信号を切断してください。入力信号をマルチメータに接続されたままでは、セルフテストが失敗します。

セルフテストには50秒かかります。セルフテストを実行するには、以下を押します。

セルフテストに合格すると、ディスプレイに次のように表示されます。

SELFTEST PASSED

セルフテストに合格した場合は、マルチメータが動作可能状態にあること、また適切な校正および自動校正を前提条件とすれば、その測定は正確なものになることを確信できます。

いずれかのテストが失敗した場合は、**ERR**インジケータが点灯し、ディスプレイには次のように表示されます。

SELFTEST FAILED

セルフテストに失敗した場合は、1つまたは複数のエラー条件が検出されています。次の「エラー・レジスタの 読み取り」のセクションを参照してください。

エラー・レジスタの読み取り

ディスプレイの**ERR**インジケータが点灯した場合には、1つまたは複数のエラーが検出されています。ハードウェア・エラーの記録は、補助エラー・レジスタに保存されます。プログラミングや構文上のエラーの記録は、エラー・レジスタに保存されます。保存されているエラー・レコードを読み取るには、以下を押します。

一番小さい番号のエラーとそのエラーの説明が表示されます。例えば、次のようなエラー・メッセージが表示されます。

209,"HARDWARE FAILURE-INTERNAL OVERLOAD:101"

メッセージ全体を表示するには、右矢印キーを使用します。エラー・メッセージに100番代のプレフィックス番号(例えば、105)が付いている場合は、プログラミングまたは構文上のエラーであることを示します。200番代のプレフィックス(例えば、209)は、ハードウェアのエラーであることを示します。

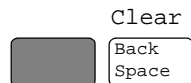
注記

ハードウェア・エラー (200番代のプレフィックス) が発生した場合は、セルフテストを再度実行してください。同じエラーが繰り返し発生する場合には、マルチメータの修理が必要な場合があります。

それでも**ERR**インジケータが点灯している場合には、さらに多くのエラーが記録されています。すべてのエラーが読み取られ、ERRインジケータが点灯しなくなるまで、上述のキー・シーケンスを繰り返します。すべてのエラーの読み取りが完了すると、エラー・インジケータはオフになります。別のエラーを読み取ろうとすると、ディスプレイに次のように表示されます。

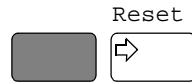
0, NO "ERROR"

エラーを検出するために、セルフテストを実行する必要はありません。本マルチメータは、ファンクションやレンジを変更した場合などに、データの入力中に発生したエラーを検出します。本器は、エラーを検出する度にビープ音を発します。

ディスプレイから情報(エラー内容など)を消去し、測定表示に戻りたい場合は、以下を押します。

注記

Back Spaceキーをシフトキーを押さずに繰り返し押すことによって、表示を消去することも可能です。

マルチメータのリセット

操作中に何度も、電源投入時の状態に戻りたくなる場合があります。マルチメータの電源を入れ直さなくても、前面パネルの**Reset**キーを押せば、電源投入時の状態に戻ります。マルチメータをリセットするには、以下を押します。

マルチメータは、図8に示されているインジケータを含むすべてのディスプレイ要素を点灯させる表示テストを実行し、リセット・プロセスを開始します。(**Reset**キーを押し続けると、マルチメータはディスプレイ・テストを連続的に実行します)。

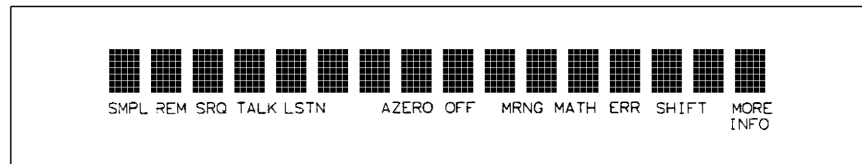
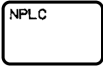
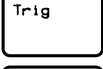
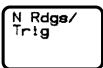
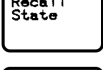



図8. ディスプレイ・テスト

注意

前面パネルのResetキーをシフトキーを押しながら押すと、マルチメータの電源を入れ直すのと同じ効果のある電源投入時シーケンスが実行されます。これによって、保存されている読み取り値や圧縮されているサブプログラムは破壊され、電源投入時SRQビットがステータス・レジスタに設定され(これらの機能については、本書で後述します)、A/Dコンバータの基準周波数がリセットされ、電源投入時セルフテストが実行されます。アルファベット・コマンド・メニュー(メニュー・キー)からRESETコマンドを実行した場合には、マルチメータは電源投入時の状態に戻りますが、電源投入時シーケンスは実行されません。メニュー・キーについては、本章で後述します。

構成キーの使用方法

構成キー(シフトキーを押さずにメニュー・キーを押す)を用いることによって、頻繁に使用されるマルチメータの機能にすばやくアクセスできます。表8には、各キー、対応するマルチメータ・コマンド、各キーの機能が示されています。(これらの機能の詳細については、第3章と第4章を参照してください)。

表8. 構成キーの機能

キー	コマンド	説明
Auto Cal	ACAL	1つまたは全ての自動校正ルーチンを実行します。(全ての自動校正ルーチンを実行するには11分かかります。自動校正を中断するために、マルチメータをリセットしないでください。自動校正を開始したら完了するまで待つ必要があります。)
NPLC	NPLC	電源ライン周波数に基づいて、積分時間を設定します。
Auto Zero	AZERO	オートゼロ機能をオン／オフします。
Offset Comp Ω	OCOMP	2端子または4端子抵抗測定のオフセット補正をオン／オフします。
Trig	TRIG	トリガ・イベントを指定します。
N Rdgs/ Trig	NRDGS	トリガ・イベントおよびサンプル・イベント当たりの読み取り回数を選択します。
Recall State	RSTATE	前に保存したステートをメモリからリコールします。
Store State	SSTATE	マルチメータの現在のステートをメモリに保存します。

Trigキーを使って、構成キーの使用方法を説明します。以下を押します。

Trig

ディスプレイには次のように表示されます。

TRIG ▦

これは、トリガ・コマンドのコマンド・ヘッダです。コマンド・ヘッダの後に自動的にスペースが入れられていることに注意してください。

パラメータの選択

多数の選択肢があるパラメータ(非数値パラメータ)については、上下の矢印キーを使って選択項目を検索できます。以下を押します。

⇩

ディスプレイには次のように表示されます。

TRIG LEVEL ▦

以下を押します。

ディスプレイには次のように表示されます。

TRIG AUTO ▦

上または下矢印キーを使用している場合、最後のパラメータ選択項目を通り過ぎると、メニューの反対側の端に戻ります。トリガを中断したい場合は、ディスプレイに次のように表示されるまで、上または下矢印キーを押し続けます。

TRIG HOLD ▦

以下を押します。

Enter

これで、トリガ・イベントは、自動(電源投入時の状態)からHOLDに変わりました。マルチメータは値の読み取りを停止します。(トリガについては、第4章で詳細に説明します)。

デフォルト値

ほとんどのパラメータがデフォルト値を持ちます。デフォルト値は、コマンドを実行する際に値を指定しなかった場合に選択される値です。例えば、トリガ・パラメータのデフォルト値はSGLです。以下を押します。

Trig

TRIG ▦

以下を押します。

Enter

本マルチメータは測定を1回実行すると停止することに注意してください(トリガが1回発生すると、以前に指定したトリガ・イベントに関係なく、そのトリガ・イベントはHOLDになります)。－1を入力してデフォルト値を選択することも可能です。以下を押します。

Enter　–　1　Enter

マルチメータはさらに、測定を1回実行した後で、停止します。

数値パラメータ

コマンドの中には数値パラメータを用いるものもあります。数値パラメータは、マルチメータによって使用される実際の値です。**NPLC**構成キーを使って、数値パラメータについて説明します。以下を押します。

NPLC

ディスプレイには次のように表示されます。

NPLC ▦

上または下矢印キーを押しても、パラメータ選択項目は表示されません。つまり、メニューがないので、数値を入力しなければなりません。例えば、以下を押します。

A/Dコンバータの積分時間として、1電源周波数を選択しました。積分時間は、A/Dコンバータが入力信号を測定する実際の時間です。(積分時間については、第3章で詳細に説明します)。

指数パラメータ

指数表現を用いて数値パラメータを入力することもできます。例えば、以下を押します。

積分時間として、0.1電源周波数を選択しました。この時点で、マルチメータをリセットし、以下を押して電源周波数を10に戻してください。

複数のパラメータ

多くのコマンドが2つ以上のパラメータを持っています。(複数のパラメータは、カンマによって区切られます)。複数のパラメータを持つコマンドの一例として、2つのパラメータを取るNRDGSコマンドを取り上げます。以下を押します。

N Rdge/
Trig

ディスプレイには次のように表示されます。

NRDGS ▦

NRDGSコマンドの最初のパラメータは、トリガ・イベント当たりの読み取り回数を指定する数値パラメータです。例えば、トリガ・イベント当たりの読み取り回数として5回を指定するには、以下を押します。

ディスプレイには次のように表示されます。

NRDGS 5, ▦

NRDGSコマンドの2番目のパラメータは、それぞれの読み取りを開始するイベントを指定します。これは数値パラメータではないので、メニューを使って指定できます。上または下矢印キーを使って、選択項目の一覧を順に検索します。ディスプレイに次のように表示されたら、

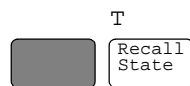
NRDGS 5,AUTO ▦

以下を押して、コマンドを実行します。

Enter

トリガ・イベント当たりの読み取り回数として5回を選択しました。例えば、TRIG SGLコマンドを実行した場合、マルチメータは読み取りを5回実行した後で、停止します。(NRDGSコマンドについては、第4章で詳細に説明します)。

メニュー・キーの使用方法

本器には、構成キーに加えて、アルファベット・コマンド・メニューがあります。このメニューへは、シフト+**C**、**E**、**L**、**N**、**R**、**S**および**T**というラベルの付いた**メニュー・**キーを用いてアクセスできます。これらの英字はそれぞれ、コマンド・メニューの領域に対応します。例えば、Tで始まるコマンドが含まれているメニューに入るには、以下を押します。

ディスプレイには次のように表示されます。

TARM ▦

これで、**メニュー・スクロール・**キー (上または下矢印キー)を使って、アルファベット順(下矢印キー)に、またはアルファベットの逆順(上矢印キー)に、メニューを順次確認することができます。例えば、上に示されているようにTARM表示から始まる場合、下矢印キーを押すことによって、ディスプレイにはアルファベット順の次のコマンド(TBUFF)が表示されます。(上または下矢印キーを押し続けることによって、メニュー内をすばやく順次確認することも可能です)。目的のコマンドを見つけたら、**Enter**キーを押して、即座に実行することができます(適用可能な場合は、デフォルトのパラメータ値を使用)。表示されているコマンドに関して、コマンド・パラメータを指定しなければならない場合には、右向きの矢印キーまたはカンマ(または、最初のパラメータが数値の場合は、任意の数値キー)を押します。これによってコマンドが選択されるので、このセクションの最初の方で説明した手順に従って、パラメータを指定/選択することができます。

使用可能なアルファベット・メニューとしては、FULLとSHORTの2種類があります。シフト+**Menu**キーを使って、これら2つのメニューの選択を切り替えることができます。指定したメニュー選択は、不揮発性メモリに保存されます(電源を切っても失われません)。FULLメニューには、コマンドに疑問符を付け加えることに

よって作ることのできる問合せコマンド(例えば、BEEP、BEEP?)を除くすべてのコマンドが含まれます。SHORTメニューは、GPIBバス関連のコマンド、前面パネルからはほとんど使用されないコマンド、専用の前面パネル・キー(例えば、**NPLC**キーや**Trig**キー)があるすべてのコマンドが削除されています。

問合せコマンド

アルファベット・コマンド・ディレクトリには、疑問符で終わるコマンドが数多くあります。これらのコマンドは、それぞれが特定の質問に対して応答するため、問合せコマンドと呼ばれます。一例として、コマンド・メニューからLINE?問合せコマンドにアクセスし、**Enter**キーを押します。マルチメータは、電源周波数を測定し、表示することによって、このコマンドに対して応答します。(応答全体を表示するには、右向き矢印キーを使用します)。もう一例として、コマンド・メニューからTEMP?コマンドにアクセスし、**Enter**キーを押します。このコマンドは、マルチメータの内部温度を摂氏温度で返します。

標準的な問合せ

FULLコマンド・メニューには、次の標準的な問合せコマンドが含まれています。

```
AUXERR?        MCOUNT?
CAL?           MSIZE?
CALNUM?        OPT?
ERR?           REV?
ERRSTR?        SSPARM?
ID?            STB?
ISCALE?        TEMP?
LINE?
```

追加問合せ

上にリストされている問合せ以外にも、マルチメータのプログラムに使用可能なコマンドに疑問符を付けることによって、問合せを作成することが可能です。例えば、AZEROコマンド(**Auto Zero**構成キー)は、オートゼロ機能をオン/オフします。AZEROコマンドに疑問符を付けることによって、現在のオートゼロ・モードを判定することができます。これを行なうには、以下を押します。

マルチメータは、現在のオートゼロ・モードを表示して応答します(power_on mode = ON)。(このコマンドは即座に実行されるので、**Enter**キーを押す必要はありません)。

注記

QFORMATコマンドを使って、問合せ応答が数値、アルファベット、アルファベットと数値の組合わせのいずれになるかを指定することができます。詳細については、第6章の「QFORMATコマンド」を参照してください。

ディスプレイの制御

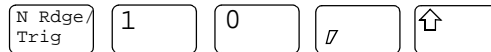
シフト+**Clear**キー、**Back Space**キーおよび**Display/Window**キー(左右の矢印キー)を使って、ディスプレイを制御することができます。

ディスプレイの消去

ディスプレイから情報(問合せ応答など)を消去したい場合には必ず、以下を押します。

ディスプレイの編集

Back Spaceキーを使って、コマンド文字列の入力中やリコール中(後述)に、コマンド文字列を部分的に編集することができます。アルファベット・パラメータやコマンド・ヘッダについては、**Back Space**キーを1回押すと、パラメータまたはヘッダ全体が消去されます。カンマ、スペース、数値パラメータについては、**Back Space**キーを1回押す度に、1文字だけ消去されます。例えば、以下を押します。

N Rdge/Trig　1　0　,　⇧

ディスプレイには次のように表示されます。

NRDGS 10,LINE ▓

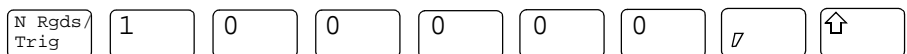
Back Spaceキーを1回押すことによって、2番目のパラメータ全体(LINE)が消去されます。ディスプレイには次のように表示されます。

NRDGS 10, ▓

ここで**Back Space**を1回押すことによって、カンマが消去されます。**Back Space**をさらに2回押すと、数字が2つとも(10)消去されます。この時点で、テンキーを使って最初のパラメータを、**メニュー・スクロール・**キーを使って2番目のパラメータをそれぞれ再入力することができます。編集したコマンドを実行するには、**Enter**キーを押します。

ディスプレイに収まらない長い文字列の表示

16文字以上のコマンドを入力する場合、それらのコマンドを入力できるように、前に入力した文字はディスプレイの左側からスクロールされて消えます。**Display/Window**キー(左右の矢印キー)を使って左または右にスクロールすることによって、行全体を表示することができます。**Display/Window**キーを使って、エラー・メッセージ、校正文字列(CALSTR?コマンド)、ユーザ定義キーの定義(後述)などの長い文字列を表示することも可能です。例えば、以下を押します。

N Rgds/Trig　1　0　0　0　0　0　,　⇧

ディスプレイに次のように表示されます。

DGS 100000,LINE ▓

左矢印キーを押すことによって、コマンドの最後の部分をディスプレイの右側からスクロールして消すことにより、最初の部分を表示することができます。今度は、右矢印キーを押すことによって、コマンドの最後の部分を表示して、最初の部分をディスプレイの左側からスクロールして消すことができます。

注記

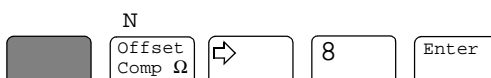
ディスプレイをウィンドウと考えれば、矢印キーを使って左右に移動することが可能です。

MORE INFO表示

Diaplay/Windowキーを使えば、ディスプレイを左右にスクロールできるだけでなく、ディスプレイの**MORE INFO**インジケータが点灯している場合に、追加ディスプレイ情報を表示できます。例えば、アルファベット・コマンド・メニューからSETACV RNDMコマンドにアクセスし、実行します。ここでは、前面パネルの**ACV**キーを押します。マルチメータの**MORE INFO**インジケータが点灯していることに注意してください。これは、表示しきれない情報があることを意味します。以下を押します。

現在のAC電圧測定法(SETACV RNDM)が表示されます。この時点で、マルチメータをリセットし、以下を押すことによって、電源投入時の状態に戻します。

表示桁数

マルチメータが表示する読み取り値の桁数を変えることができます。電源投入時の状態では、マルチメータの分解能は8.5桁ですが、ディスプレイの表示桁数は7.5桁です。8.5桁すべてを表示するには、以下を押します。

注記

表示桁数を指定している場合、ディスプレイの最左端の桁(1/2桁と呼ばれる)が含まれます。

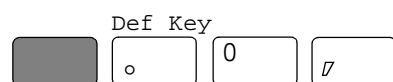
NDlGコマンドは、ディスプレイから桁をマスクするだけです。読み取り値メモリに送信される読み取り値や、GPIBバス経由で転送される読み取り値には影響を及ぼしません。また、マルチメータの分解能を上回る桁数を表示することはできません。

リコール

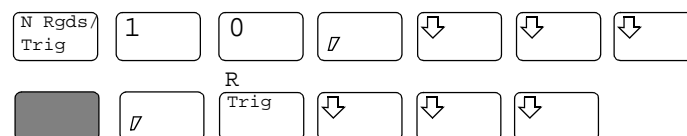
コマンド入力プロセスを繰り返さなくても、最後に実行したコマンドを簡単にリコールできます。以下を押します。

ディスプレイには、最後に実行されたコマンドが表示されます。(**Reset**や**DCV**などの即座に実行されるコマンド、または校正保護コードが含まれたコマンドをリコールすることはできません。上のキーボード入力を繰り返すことによって、以前に実行したコマンドをリコールすることが可能です。所望のコマンドをリコールしたら、コマンドを変更し(このセクションで前述の「ディスプレイの編集」を参照)、**Enter**を押して実行します。

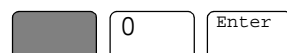
ユーザ定義キー

1つまたは複数のコマンドから構成される文字列を、**f0～f9**のラベルが付けられた**ユーザ・**キーの各々に割り当てることができます。これらのキーのうちの1つに文字列を割り当てたら(最大文字列長は40文字)、該当するキーを押して、ディスプレイ上に文字列を表示します。次に、**Enter**キーを押すことによって、文字列を実行できます。**Defキー**を使えば、任意のユーザ定義キーにコマンド文字列を割り当てることができます。例えば、コマンドNRDGS 10, AUTO; TRIG SGL (セミコロンは複数のコマンドをリンク)をユーザ定義キー **f0**に割り当てるには、以下を押します。

ディスプレイには次のように表示されます。

DEFKEY 0,"

以下を押すことによって、コマンド文字列を入力できます。

文字列を保存するには(これによって文字列が実行されることはなく、単に文字列がユーザ定義キーに割り当てられる)、以下を押します。

Enter

キー**f0**に割り当てられている文字列にアクセスし、実行するには、以下を押します。

マルチメータは、測定を10回実行した後で、停止します。

キーボードの特殊な機能の1つとして、シフトキーを押さずに(ただし、コマンドの入力中は除く)、キーに割り当てられている文字列にアクセスできます。例えば、以下を押すことによって、**f0**に割り当てられている文字列にアクセスし、実行することができます。

コマンド・メニューからユーザ定義キーにコマンドを割り当てることも可能です。即時実行キー (DCV、ACVなど)を使ってコマンドを割り当てることはできません。その代わり、コマンドへはメニューからアクセスしなければなりません。前面パネルから保存したキー定義は、前面パネルから編集することができます。(コントロー

ラからダウンロードしたキー定義を編集することはできません)。編集を行なうには、該当するユーザ定義キーを押し、文字列が表示されている間に文字列を編集します(このセクションの前述の「ディスプレイの編集」を参照)。文字列の編集が済んだら、**Enter**キーを押して文字列を実行します。(前の文字列は依然としてそのユーザ定義キーに割り当てられています)。編集した文字列をユーザ定義キーに再割り当てすることはできません。キー定義を変更したい場合には、上のステップを繰り返えしてください。

キーボード・オーバレイの取付け

図9は、ユーザ・キーの表面を完全に覆うキーボード・オーバレイを示したものです。このオーバレイには、各ユーザ定義キーに割り当てられているコマンドをペンを使って明記することができます。

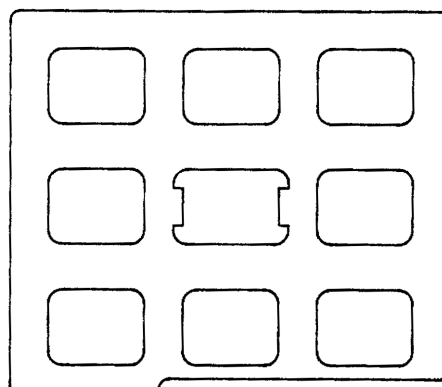



図9. キーボード・オーバレイ(Agilent部品番号03458-84303)

オーバレイは、数値キー 5の回りの鍔にオーバレイを留める2つのタブによって、しっかりと固定します。オーバレイを取り付けるには、オーバレイの左タブを鍔の左側に挿入します。図10に示されているように、オーバレイを曲げて、右側タブを鍔に押し込みます。

図10. キーボード・オーバレイの取付け

リモートからの操作

このセクションでは、マルチメータをリモートから操作するための基本事項について説明します。これには、GPIBアドレスの読み取りおよび変更、マルチメータへのコマンドの送信、マルチメータからのデータの検索が含まれます。

入出力ステートメント

リモートからマルチメータを操作するのに用いられるステートメントは、コンピュータとコンピュータ言語によって異なります。特に、コンピュータが情報の入出力に用いるステートメントを認識しておく必要があります。例えば、Hewlett-Packard シリーズ200/300 BASIC言語の入力ステートメントは以下の通りです。

ENTER または TRANSFER

出力ステートメントは以下の通りです。

OUTPUT

ご使用のコンピュータのマニュアルをお読みになり、使用する必要のあるステートメントを確認してください。本書の中で取り上げられている各例では、Hewlett-Packard シリーズ200/300 BASIC言語を使用しています。

GPIBアドレスの読み取り

リモートからマルチメータを操作する前に、マルチメータのGPIBアドレス(出荷時設定 = 22) を確認する必要があります。GPIBアドレスを確認するには、以下を押します。

通常は次のように表示されます。

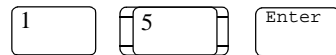
ADDRESS 22 ▦

表示されている応答が機器アドレスです。リモート・コマンドを送信する場合は、このアドレスをGPIBインタフェースのセレクト・コード(通常は7)に付加します。例えば、セレクト・コードが7で、機器アドレスが22の場合は、それらを組み合わせて722になります。

GPIBアドレスの変更

GPIBバス上のデバイスはすべて、固有のアドレスを持たなければなりません。マルチメータのアドレスを変更する必要がある場合は、コマンド・メニュー(**メニュー・キー**)からADDRESSコマンドにアクセスします。ディスプレイには次のように表示されます。

ADDRESS ▦

新しいアドレスを入力することができます。例えば、以下を押します。

アドレスを15に変更しました。アドレスを22に変更し直したい場合は、上の手順を繰り返し(または**Recall**キーを使用し)、15に代えて22と指定します。

リモート・コマンドの送信

マルチメータにリモート・コマンドを送信するには、コンピュータの出力ステートメントと、GPIBセレクト・コード、機器アドレス、最後にマルチメータ・コマンドを結合します。例えば、マルチメータにビープ音を出させるには、以下を送信します。

```
OUTPUT 722; "BEEP"
```

ディスプレイの**REM**と**LSTN**の2つのインジケータが点灯していることに注意してください。これは、マルチメータがリモート・モードにあり、リスンする(コマンドを受け取る)ようにアドレス指定されていることを意味します。

マルチメータからのデータの読み取り

本器には、読み取り値や応答を問合せコマンドに出力する機能があります。一例として、以下を送信することによって、マルチメータに問合せコマンドに対する応答を生成させます。

```
OUTPUT 722;"ID?"
```

リモートから問合せを送信した場合は、前面パネルからコマンドを実行した場合とは違って、応答は表示されません。その代わりに、マルチメータは出力バッファに応答を送ります。出力バッファは、問合せ応答または単一の読み取り値を、コンピュータによって読み取られるまで、または新しい情報に置き換えられるまで保持するレジスタです。コンピュータの入力ステートメントを使って、出力バッファから応答を取り出します。例えば、次のプログラムは、応答(HP 3458A)を読み取り、プリントします。

```
10 ENTER 722;A$
20 PRINT A$
30 END
```

同じ手法で、マルチメータから読み取り値を取り出すことができます。マルチメータの測定中に読み取り値メモリをオンにできなかった場合には常に(読み取り値メモリについては、第4章で説明します)、次のプログラムを実行することによって、読み取り値を取り出すことができます。

```
10 ENTER 722;A
20 PRINT A
30 END
```

Localキー

リモートからの操作中にマルチメータのキーボード上のキーを押した場合、マルチメータは応答しません。これは、ディスプレイの**REM**インジケータによって示されているように、マルチメータがリモート・モードにあり、**Local**キー以外はすべて無視しているためです。マルチメータをローカル・モードに戻すには、以下を押します。

Local

第3章　測定のための構成

第3章　測定のための構成

概要

本章では、ディジタイズを除くすべての種類の測定を実現できるように本器を設定するための方法について説明します。[1] 本章ではまた、サブプログラム/ステート・メモリ、入力バッファおよびステータス・レジスタの使用方法についても説明します。本章に従ってアプリケーションに合わせてマルチメータを設定したら、第4章で、測定をトリガし、読み取り値を読み取り値メモリまたはGPIB出力バッファに転送するための方法を学びます。本章では主に、以下の項目について説明します。

- 一般的な構成
- DC/抵抗測定用の構成
- AC測定用の構成
- 比測定用の構成
- サブプログラム・メモリの使用方法
- ステート・メモリの使用方法
- 入力バッファの使用方法
- ステータス・レジスタの使用方法

一般的な構成

このセクションでは、マルチメータのセルフテスト、校正要件、およびほとんどの測定ファンクションに適用される一般的な構成について説明します。

セルフテスト

測定用に構成する前に、セルフテストを実行して、マルチメータが動作可能状態にあることを確認する必要があります。セルフテストが完了するまでには、約50秒かかります。セルフテストを実行するには、以下を送信します。

```
OUTPUT 722;"TEST"
```

セルフテストに合格した場合は、マルチメータが動作可能状態にあること、また適切な校正を前提条件とすれば、その測定は正確なものになることを確信できます。1つまたは複数のテストが失敗した場合は、マルチメータはビットを補助エラー・レジスタに設定します。補助エラー・レジスタはまた、エラー・レジスタのビット0を設定し、ディスプレイの**ERR**インジケータが点灯します。

---

1. この章では、特にディジタイズについて取り扱っていませんが、「一般的な構成」のほとんどの部分がディジタイズに適用できます。ディジタイズに関する固有の情報は、第5章を参照していください。

エラー・レジスタの読み取り

ハードウェア・エラーが検出された場合には、マルチメータはビットを補助エラー・レジスタに設定し、ビット0もエラー・レジスタに設定します。プログラミング・エラーが検出された場合には、マルチメータはビットをエラー・レジスタにだけ設定します。

ERRSTR?コマンドは、各エラーを(一度に1つずつ)読み取り、対応するビットをクリアします。1つ以上のビットが補助エラー・レジスタに設定されている場合には、ERRSTR?コマンドは、エラー・レジスタに進む前にまず、補助エラー・レジスタを読みます。ERRSTR?コマンドは2つの応答を返します。最初の応答は、最下位(最も値の小さい)設定ビットの値(10進数)です。2番目の応答は、エラーを説明するメッセージ(文字列)です(返される文字列の最大長は200文字です)。ビットの読み取りが終わると、ERRSTR?コマンドはそのビットをクリアします。

次のプログラムでは、ERRSTR?コマンドを使って、すべてのエラーを一度に1つずつ読み取っています。すべての設定ビットが読み取られ、クリアされると、すなわち、いずれのレジスタにも設定ビットが存在していない場合は、ERRSTR?コマンドは0、すなわち「エラーなし」を返します。

```
10 OPTION BASE 1                    !配列の番号付けは1から開始する
20 DIM A$[200]                      !文字列配列を宣言する
30 OUTPUT 722; "ERRSTR?"            !エラー・メッセージを読み取る
40 ENTER 722; A,A$                  !数値をAに、文字列をA$に入力する
50 PRINT A,A$                       !応答をプリントする
60 IF A>0 THEN GOTO 30              !ループして各エラーを読み取る
70 END
```

ERR?およびAUXERR?コマンドは、エラー・レジスタと補助エラー・レジスタ内の全設定ビットの十進和をそれぞれ返します。サンプル・プログラムと発生する可能性のあるエラーのリストについては、第6章の各コマンドの説明を参照してください。

校正

本器では、外部校正と自動校正の2通りの校正が可能です。外部校正には、外部基準信号源を使った手順が含まれます。外部校正の詳細については、『3458校正マニュアル』を参照してください。

CALNUM?問合せコマンドは、マルチメータの外部校正が行なわれた回数を示す数を返します。この数を日常的にチェックすることによって、マルチメータで実行された校正をモニタすることができます。次のプログラムは、現在の校正回数を読み取り、返します。

```
10 OUTPUT 722;"CALNUM?"
20 ENTER 722;A
30 PRINT A
40 END
```

自動校正

本器には、DCV、AC、OHMS、ALLの4種類の自動校正ルーチンが装備されています。これらのルーチンは、ほとんどの測定ファンクションの確度を短期的に向上させますが、本器の定期的な外部校正に取って代わるものではありません。各ルーチンの影響を受ける測定ファンクションは以下の通りです。

- DCVルーチンは、全測定ファンクションを向上させます。このルーチンが完了するまでには、約1分かかります。

- ACルーチンは、ACまたはAC+DC電圧(すべての測定法)、ACまたはAC+DC電流、直接またはサブ・サンプリングしたディジタイズ(ACまたはDC結合)、周波数および周期測定用の特定のファンクションを向上させます。ACルーチンが完了するまでには、約1分かかります。

- OHMSルーチンは、2端子または4端子抵抗測定、DC電流およびAC電流測定用の特定のファンクションを向上させます。OHMSルーチンが完了するまでには、約10分かかります。

- ALLルーチンは、上のルーチンをすべて実行することによって、すべての測定ファンクションを向上させます。ALLルーチンが完了するまでには、約11分かかります。

注記 自動校正の実行中は、マルチメータの電源を入れ直したり、リセットしたりしないでください。もしそうした場合は、自動校正の定数のほとんどが消去されるため、マルチメータはACAL REQUIREDエラーを生成します。この場合には、ALLルーチンを実行して、エラーを取り除く必要があります。

DCVルーチンはすべての測定ファンクションに適用されるため、ACまたはOHMS自動校正を実行する前にDCV自動校正を実行するか、全ルーチン(ALL)を実行します(以下に示されている2番目の例を参照してください)。

自動校正の実行

4端子抵抗測定を実行しようとしていると仮定します。DCV自動校正ルーチンはあらゆる測定の短期的な確度を、OHMS自動校正は抵抗測定(および電流測定)を向上させます。次のプログラムは、DCV自動校正に続けて、OHMS自動校正を実行します。

```
10 OUTPUT 722; "ACAL DCV"
20 OUTPUT 722; "ACAL OHMS"
30 END
```

自動校正にセキュリティがかけられている場合は(工場出荷時には自動校正にセキュリティがかけられていません)、自動校正を実行するためには、セキュリティ・コードを入力しなければなりません。詳細については、第6章の「ACALコマンド」を参照してください。以下を送信することによって、自動校正ルーチンをすべて(まずDCV、次にOHMSおよびAC)実行できます。

```
OUTPUT 722; "ACAL ALL"
```

自動校正を実行する前に、必ず、すべての入力信号を切断してください。入力信号をマルチメータに接続されたままにした場合は、自動校正と後続の測定に悪影響が及ぶ恐れがあります。

自動校正を使用すべき場合

確度を最大限に高めるため、24時間に1度、またはマルチメータの温度が、直近の外部校正が実行されてからまたは直近の自動校正が実行されてから、±1°C変化した場合に、ACAL ALLを実行することをお勧めします。(CALSTRコマンドを使って、キャリブレータにマルチメータの内部校正温度を保存させることをお勧めします。保存した温度は、後でCALSTR?コマンドを使って読み取ることができます)。

次の例は、TEMP?コマンドを使ったマルチメータの内部温度(摂氏温度)のモニタ方法を示したものです。

```
10 OUTPUT 722;"TEMP?"
20 ENTER 722;A
30 PRINT A
40 END
```

自動校正の定数は、不揮発性メモリに保存されます(電源を切っても影響を受けません)。したがって、電源を入れ直したという理由だけで、自動校正を実行する必要はありません。

入力端子の選択

本器には、測定接続用の端子が前面と裏面の両方にあります。前面パネルの**Terminals**スイッチを使って、2つの端子の選択を切り替えることができます(押し下げられている場合 = **裏面**、上がっている場合 = **前面**)。入力端子をリモートから選択することはできません。本章の測定用接続図には、前面端子の接続だけが示されています。裏面端子の接続については、各端子を同様にラベルの付けられた裏面端子に接続します。すべての測定接続に、高インピーダンス、低誘電吸収のケーブルを用いることをお勧めします。

警告　必ず、危険を認識している資格のある修理技術者だけが、マルチメータの取り外しや設置、マルチメータへの配線を行なってください。カバーを取り外したり、電源電圧の切替えスイッチを変更したり、電源ヒューズを取付けたり交換したりする場合には、マルチメータの電源コードを抜いてから行なってください。

高電圧測定には常に危険を伴います。42Vを超える入力がいずれかの端子に接続されている場合には必ず、マルチメータのすべての入力端子(前面と裏面の両方)が危険であると認識する必要があります。いずれかの端子に最大電圧が印加された場合は、すべての端子がそれと同じ電位差であると考えてください。

配線作業者は、人体に危険な電圧を持つ配線すべてにラベルを貼り付けることをお勧めします。このラベルは、できるだけ入力端子の近くに貼り付け、赤や黄色などの人目を引く色にしてください。ラベルには、高電圧の可能性があることを明確に示してください。

注意　電流入力端子(I)は、±1.5Aピーク、最大非破壊入力は、<1.25A RMSです。電流入力はヒューズで保護されています。マルチメータの入力電圧定格は以下の通りです。

表9. 入力定格

	定格入力	最大非破壊入力
HIからLOInput:	± l000Vピーク	±1200Vピーク
HI/LO Ω SenseからLO Input:	± 200Vピーク	± 350Vピーク
HIからLO Ω Sense:Input:	± 200Vピーク	± 350Vピーク
LO InputからGuard:	± 200Vピーク	± 350Vピーク
Guardからグランド:	± 500Vピーク	± 1000Vピーク
HI/LO Input、HI/LO Ω Sense、またはI端子からグランド:	± l000Vピーク	± 1500Vピーク
前面端子から裏面端子:	± l000Vピーク	± l500Vピーク

上の最大非破壊入力のいずれかが超過した場合には、マルチメータに損傷が及びます。

ガード

本章の測定接続図には、測定信号源のロー側に接続されたマルチメータの**Guard**端子(ガード測定)が示されています。この構成では、GuardスイッチがOpen(上がっている状態)位置にあるという仮定のもとで、最大の**実効コモン・モード・ノイズ除去比**(ECMR)が、**Terminals**スイッチによって選択された入力端子で実現されます。非ガード測定を行なう場合は、**Guard**スイッチを押し下げ(**TO LO**位置)、**Guard**端子を測定信号源に接続しません。**TO LO**位置では、**Guard**スイッチは、**Terminals**スイッ
チによって選択された端子の**LO Input**端子に、**Guard**端子を内部接続します。この構成では、ECMRが減少します。付録Aの仕様は、ガード測定用のECMRを示したものです。すべての測定接続に、高インピーダンス、低誘電吸収のケーブルを用いることをお勧めします。

読み取りの停止

マルチメータの電源投入時の状態では、トリガ・アーム、トリガおよびサンプリング・イベントはAUTOに設定されています(これらのイベントについては、第4章で詳細に説明します)。このため、マルチメータは、連続的に読み取りを行います。マルチメータを測定用に構成する前に、読み取りを停止する必要があります。読み取りを停止することによって、構成に要する時間が短縮されるだけでなく、不要な読み取り値が読み取り値メモリまたはGPIB出力バッファに保存される可能性がなくなります。マルチメータを初期設定するか(次項を参照)、トリガ・アームまたはトリガ・イベントを次のようにHOLDに設定することによって、読み取りを停止することができます。

```
OUTPUT 722; "TARM HOLD"
```

または

```
OUTPUT 722; "TRIG HOLD"
```

マルチメータの構成が済んだら、トリガ・アームまたはトリガ・イベントをHOLDから何か他のイベントに変更することによって、測定をオンにすることができます。(測定のトリガの詳細については、第2章を参照してください)。

マルチメータの初期設定

PRESET NORMコマンドは、RESETコマンドと似ていますが、リモート操作に適した測定開始点となるようにマルチメータを構成します。(RESETは主として、前面パネル用です)。PRESET NORMは、マルチメータを既知の構成に設定し、トリガ・イベントを同期(TRIG SYN)コマンドに設定することによって読み取りを一時停止するため、マルチメータを構成する際の最初のステップとして実行するのが良いでしょう。表10には、PRESET NORMコマンドによって実行されるコマンドが示されています。

表10. PRESET NORMの状態

コマンド	説明
ACBAND 20,2E+6	AC帯域幅20Hz～2MHz
AZERO ON	オートゼロをオンにする
BEEP ON	警報器をオンにする
DCV AUTO	DC電圧測定、オートレンジ
DELAY –1	デフォルトの遅延
DISP ON	表示をオンにする
FIXEDZ OFF	固定入力抵抗をオフにする
FSOURCE ACV	周波数／周期信号源はAC電圧である
INBUF OFF	入力バッファをオフにする
LOCK OFF	キーボードをオンにする
MATH OFF	リアルタイム演算をオフにする
MEM OFF	読み取り値メモリをオフにする
MFORMAT SREAL	単精度実数読み取り値メモリ・フォーマット
MMATH OFF	後処理演算をオフにする
NDIG 6	6.5桁を表示する
NPLC 1	1電源周波数の積分時間
NRDGS 1,AUTO	トリガ当たりの読み取り回数1回、オート・サンプリング・イベント
OCOMP OFF	オフセット補正抵抗をオフにする
OFORMAT ASCII	SCII出力フォーマット
TARM AUTO	オート・トリガ・アーム・イベント
TIMER 1	タイマ間隔1秒
TRIG SYN	同期トリガ・イベント

以下を除くすべての演算レジスタが0に設定される:
DEGREE = 20
PERC = 1
REF = 1
RES = 50
SCALE = 1

リモートから初期設定しようとした場合に、マルチメータがビジーであるか、GPIBインタフェースがホールド状態にある可能性があります。いずれの場合も、マルチメータはリモート・コマンドに応答しません。マルチメータを初期設定する前に、GPIB Device Clearコマンドを送信すると良いでしょう。マルチメータは、Device Clearコマンドに即座に応答します。次のプログラムは、Device Clearコマンドを送信した後で、PRESET NORMコマンドを送信します。

```
10 CLEAR 722
20 OUTPUT 722;"PRESET NORM"
30 END
```

本器は、PRESET NORMコマンドに加えて、第4章で説明するPRESET FASTコマンド(高速測定および転送用に構成)、第5章で説明するPRESET DIGコマンド(DCVディジタイズ用に構成)を備えています。

測定ファンクションの指定

FUNCコマンドの最初のパラメータは、測定ファンクションを選択します。例えば、DC電圧測定を指定するには、以下を送信します。

```
OUTPUT 722;"FUNC DCV"
```

TFUNCコマンド・ヘッダはオプションなので、省略できます。例えば、以下を送信することによって、DC電圧測定を指定することが可能です。

```
OUTPUT 722;"DCV"
```

本章の残りの例では、短縮(FUNCヘッダなし)バージョンを使用しています。表11には、各種測定ファンクション・パラメータと、各パラメータによって選択されるファンクションが示されています。

表11. 測定ファンクション・パラメータ

ファンクション・パラメータ	説明
ACDCI	AC電流測定を選択する、DC結合
ACDCV	AC電圧測定を選択する、DC結合
ACI	AC電流測定を選択する、AC結合
ACV	AC電圧測定を選択する、AC結合
DCI	DC電流測定を選択する
DCV	DC電圧測定を選択する
DSAC *	直接サンプリング、AC結合
DSDC *	直接サンプリング、DC結合
FREQ	周波数測定を選択する
OHM	2端子抵抗測定を選択する
OHMF	4端子抵抗測定を選択する
PER	周期測定を選択する
SSAC *	サブ・サンプリング、AC結合
SSDC *	サブ・サンプリング、DC結合

* これらのファンクションの詳細については、第5章「ディジタイジング」を参照してください。

オートレンジ

オートレンジ・ファンクションをオンにした場合、マルチメータは読み取りを行なう前に必ず入力をサンプリングし(読み取りがトリガされている場合)、適切なレンジを自動的に選択します。オートレンジには入力のサンプリングが必要なため、オートレンジをオンにした状態で行なわれる測定は、固定レンジで行なわれる測定よりも時間がかかります。電源投入時/PRESET NORM状態では、オートレンジはオンになっています。非常に安定した入力信号を測定する場合は、ARANGE ONCEコマンドを使ってオートレンジによる適切なレンジの選択を可能にした後で(読み取りがトリガされている場合)、後続の読み取りについてはオートレンジをオフにします。これによって、オートレンジの利点である自動選択だけでなく、オートレンジをオフのした場合の読み取りの高速性という利点も得られます。これを行なうためには、以下を送信します。

```
OUTPUT 722;"ARANGE ONCE"
```

これで、トリガが開始した場合、マルチメータは適切なレンジを選択した後で、オートレンジをオフにします。後でオートレンジをオンにしなければならなくなった場合には、以下を送信します。

```
OUTPUT 722; "ARANGE ON"
```

レンジの指定

ファンクション・コマンド(ACV、DCV、OHMなど) またはRANGEコマンドのうちの1つの最初のパラメータを使って、固定レンジを指定します。このパラメータは、入力信号の最大予想振幅(または抵抗測定の場合は最大抵抗)として指定するため、*max._input*と呼ばれます。マルチメータは適切なレンジを選択します。*max._input*を指定する場合は、入力信号の絶対値を使用します(負の数ではない)。例えば、最大入力－2.5 VのDC電圧を指定するには、以下を送信します。

```
OUTPUT 722; "DCV 2.5"
```

この場合、10 VDCレンジが選択されます。測定ファンクションを変更せずに、別の*max._input* (例えば、15V)を指定するには、以下を送信します。

```
OUTPUT 722; "RANGE 15"
```

この場合、100Vレンジが選択されます。

注記　周波数測定や周期測定の場合は、*max._input* パラメータは入力信号の最大振幅を指定します。このパラメータは、周波数レンジ(Hz)や周期レンジ(秒)は指定しません。

*max._input*パラメータを省略するか、AUTOを指定することによって、オートレンジ・モードを選択します。例えば、DCVコマンドを使ってオートレンジを選択するには、以下を送信します。

```
OUTPUT 722; "DCV"
```

各測定ファンクションのレンジについては、第6章の「FUNCコマンド」または「RANGEコマンド」を参照してください。

DC/抵抗測定用の構成

このセクションでは、DC電圧、DC電流、2端子/4端子抵抗(Ω)測定を実行するためのマルチメータの構成方法について説明します。

DC電圧

本器は、5種類のうちのいずれかのレンジでDC電圧を測定します。表12は、各DC電圧と、そのフルスケール読み取り値を示したものです(そのレンジの最高桁数も示されています)。表12には、各レンジの最高分解能と入力抵抗も示されています。(分解能は、指定した積分時間の関数です。詳細については、本章で後述の「積分時間の設定」を参照してください)。図11は、あらゆる種類の電圧測定の前面端子の接続を示したものです。電源投入時/PRESET NORM状態においては、DC電圧測定が選択されています。DCVコマンドを使ってDC電圧測定を指定することも可能です。例えば、1Vレンジでの DC電圧測定を指定するには、以下を送信します。

```
OUTPUT 722;"DCV 1"
```

表12. DC電圧レンジ

DCVレンジ	フルスケール読み取り値	最高分解能	入力抵抗
100mV	120.00000mV	10nV	>10GΩ*
1V	1.20000000V	10nV	>lOGΩ*
10V	12.0000000V	100nV	>lOGΩ*
100V	120.000000V	1µV	10MΩ
1000V	1050.00000V	10µV	1OMΩ

* FIXEDZ OFFでも、FIXEDZ ONでも、入力抵抗は10MΩで固定です。
詳細については、本章後述の「固定入力抵抗」を参照してください。

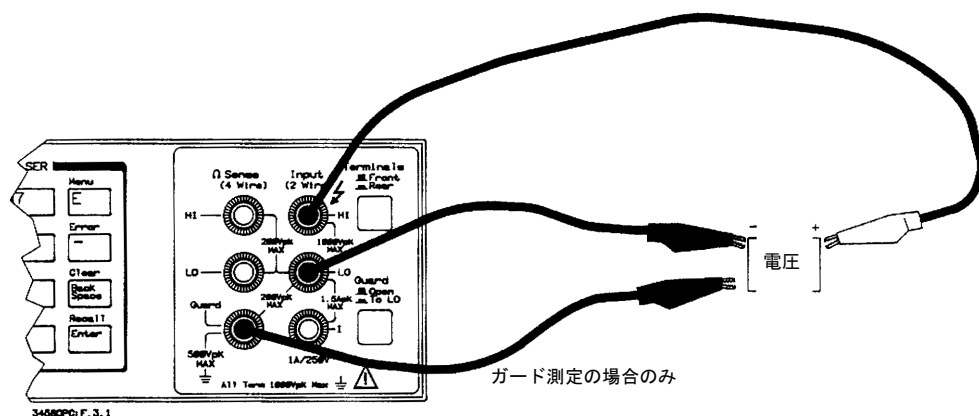



図11. 電圧測定用の接続

DC電流

本器は、入力端子間に内部シャント抵抗を入れ、抵抗に印加された電圧を測定し、電流を計算する(電流 = 電圧/抵抗)ことにより、電流を測定します。本器の前面と裏面にある電流入力は、1 A、250Vヒューズによって保護されています。図12は、あらゆる種類の電流測定の前面端子の接続を示したものです。

本器は、8種類のうちのいずれかのレンジでDC電流を測定します。表13は、各DC電流と、そのフルスケール読み取り値を示したものです(フルスケール読み取り値には、各レンジの最高桁数も示されています)。表13には、各レンジの最高分解能と適用されるシャント抵抗も示されています。(分解能は、指定した積分時間の関数です。詳細については、本章で後述の「積分時間の設定」を参照してください)。DCIコマンドを使ってDC電流測定を指定します。例えば、10µAレンジでのDC電流測定を指定するには、以下を送信します。

```
OUTPUT 722;"DCI 10E-6"
```

表13. DC電流レンジ

DCIレンジ	フルスケール読み取り値	最高分解能	シャント抵抗
lOOnA	120.000nA	1pA	545.2kΩ
1µA	1.200000µA	1pA	45.2kΩ
10µA	12.000000µA	1pA	5.2kΩ
100µA	120.00000µA	10pA	730Ω
1mA	1.2000000mA	lOOpA	100Ω
10mA	12.000000mA	1nA	10Ω
100mA	120.00000mA	10nA	1Ω
1A	1.0500000A	100nA	0.1Ω

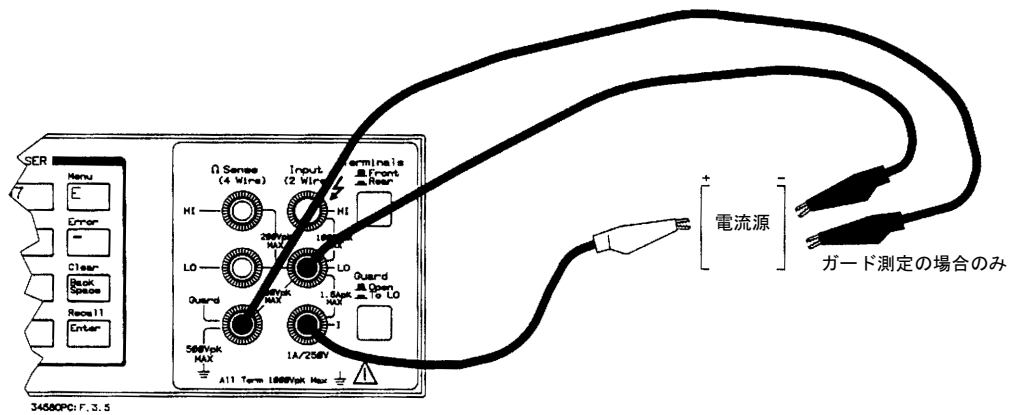



図12. 電流測定用の接続

抵抗

本器は、測定する未知の抵抗に既知の電流を流すことによって、抵抗を測定します。抵抗に流れる電流によって抵抗全体に電圧が生じます。本器は、この電圧を測定し、未知の抵抗を計算します(抵抗 = 電圧/電流)。表14は、2端子抵抗測定と4端子抵抗測定のレンジと、そのフルスケール読み取り値を示したものです(フルスケール読み取り値には、各レンジの最高桁数も示されています)。表14には、各レンジの最高分解能と測定電流も示されています。(分解能は、指定した積分時間の関数です。詳細については、本章で後述の「積分時間の設定」を参照してください)。

表14. 抵抗レンジ

抵抗(F)レンジ	フルスケール読み取り値	最高分解能	測定電流
10Ω	12.00000Ω	10µΩ	10mA
100Ω	120.00000Ω	10µΩ	1mA
1kΩ	1.2000000kΩ	100µΩ	1mA
10kΩ	12.000000kΩ	1mΩ	l00µA
100kΩ	120.00000kΩ	10mΩ	50µA
1MΩ	1.2000000MΩ	100mΩ	5µA
10MΩ	12.000000MΩ	1Ω	500nA
100MΩ	120.00000MΩ	10Ω	500nA
1GΩ	1.2000000GΩ	100Ω	500nA

2端子抵抗測定

2端子抵抗測定は、試験用リード線の抵抗が測定値をはるかに下回る場合に最もよく用いられます。リード線抵抗が測定抵抗に比べて大きい場合は、読み取りが不正確になります。例えば、3 m近く離れた場所の1Ω抵抗を測定すると仮定します。24ゲージの銅線を使って接続した場合、約6 mのリード線は、測定に対して、約0.5Ωの影響を与えます。このため、全測定抵抗は1.5Ω、誤差は50%となります。大きなリード抵抗が生じる可能性のあるこの他の要因としては、接続のゆるみや汚れ、ワイヤのよじれや損傷、超高温環境などが挙げられます。NULL演算を行なうことによって、2端子抵抗測定の確度の向上を図ることができます(詳細については、第4章の「ヌル」を参照してください)。図13は、2端子抵抗測定用の前面パネルでの接続を示したものです。OHMコマンドを使って2端子抵抗測定を指定します。例えば、lkΩレンジでの2端子抵抗測定を指定するには、以下を送信します。

```
OUTPUT 722;"OHM 1E3"
```

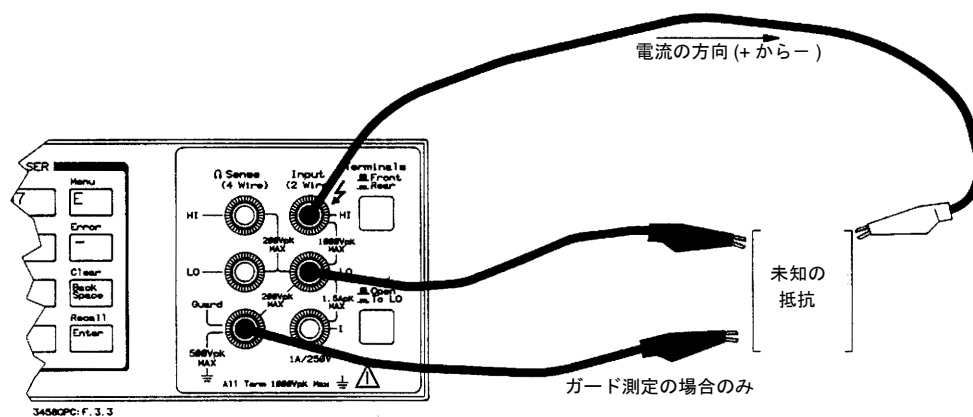



図13. 2端子抵抗測定用の接続

4端子抵抗測定

4端子抵抗測定モードでは、試験用リード線の抵抗に起因する測定誤差がなくなります。2端子抵抗測定では、試験用リード線の抵抗と未知の抵抗の合成抵抗で電圧測定が行なわれます。4端子抵抗測定では、合成抵抗ではなく、未知の抵抗のみの電圧が測定されます。これは、試験用リード線の抵抗が測定抵抗に比べて大きい場合に重要です。図14は、4端子抵抗測定用の前面部の接続を示したものです。OHMFコマンドを使って4端子抵抗測定を指定します。例えば、10MΩレンジでの4端子抵抗測定を指定するには、以下を送信します。

```
OUTPUT 722; "OHMF 10E6"
```

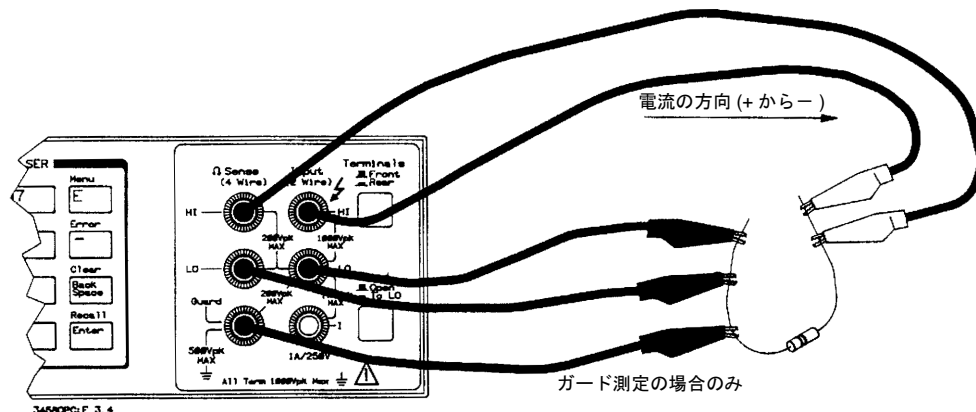



図14. 4端子抵抗測定用の接続

A/Dコンバータの構成

A/Dコンバータの構成によって、DC/抵抗測定の測定速度、分解能、確度およびノーマル・モード・ノイズ除去[1] が決まります。A/Dコンバータの構成に影響を及ぼす要因としては、基準周波数、指定した積分時間、指定した分解能が挙げられます。

基準周波数

電源を入れると、マルチメータは電源周波数を測定し、その値を50 Hzまたは60 Hzに丸め、A/Dコンバータの**基準**周波数を丸められた値に設定します。(電源周波数が400Hzの場合は、**基準周波数**として、400Hzの分周波である50Hzが用いられます)。DC/抵抗測定については、積分時間が≧1電源周波数である場合に、マルチメータはノーマル・モード・ノイズ除去(NMR)を基準周波数で実現します。詳細については、後の「積分時間の設定」のセクションを参照してください。

基準周波数の変更

ほとんどの動作条件では、電源投入時の基準周波数で、優れたNMRを実現します。ただし、最高のNMRを実現するためには、基準周波数を正確に電源周波数に設定する必要があります。(電源周波数がドリフトしやすい場合には、基準周波数を定期的に補正する必要があります)。次のコマンドは、電源周波数を測定し、測定した値と正確に同じ基準周波数を設定します(電源周波数が400Hzの場合、マルチメータは測定値を8で割った値を基準周波数として使用します)。

```
OUTPUT 722;"LFREQ LINE”
```

LFREQコマンドを使って、基準周波数を直接指定することも可能です。これは、マルチメータが被測定デバイスとは異なる電源周波数を持つ場合に、特に有用です。例えば、マルチメータの電源周波数が60 Hz で、被測定デバイスの電源周波数が50 Hzであると仮定します。この場合は、次のように基準周波数を50 Hz に設定することによって、NMRを実現することができます。

---

1. ノーマル・モード・ノイズ除去(NMR)とは、DCまたは抵抗測定において、電源周波数のノイズを除去するマルチメータの能力です。

```
OUTPUT 722;"LFREQ 50"
```

電源を入れ直すか、前面パネルの**Reset**キーを押す度に、基準周波数は50 Hzか60 Hzの丸められた値になることに注意していください。

積分時間の設定

積分時間とは、A/Dコンバータが入力信号を測定する期間のことを言います。DC/抵抗測定については、積分時間によって、測定速度、確度、分解能の最高桁数、A/Dコンバータの基準周波数におけるノーマル・モード・ノイズ除去比が決まります。積分時間は、NPLCコマンドを使って電源周波数(PLC)に換算して指定するか、APERコマンドを使って直接(秒単位で)指定することができます。NPLCコマンドとAPERコマンドはともに、積分時間を設定するため、いずれかのコマンドを実行すると、もう一方のコマンドによって以前に設定された積分時間がキャンセルされます。

電源周波数の指定

マルチメータは、積分時間が≧1電源周波数である場合に、ノーマル・モード・ノイズ除去(NMR)をA/Dコンバータの基準周波数で実現します。積分時間は、NPLCコマンドを使って電源周波数(PLC)に換算して指定することが可能です。マルチメータは、指定されたPLCにA/Dコンバータの基準周波数の周期を乗算して(LFREQコマンド)、積分時間を決定します。例えば、50 Hz電源の周期はl/50 = 20 msになります。10 PLCを指定した場合、積分時間は200 msになります。電源投入時の状態では、積分時間は10 PLCに設定されています。PRESET NORM状態では、積分時間は1 PLCに設定されています。最低確度、最低分解能、NMRなしでの最高速測定を実現するための積分時間を設定するには、以下を送信します。

```
OUTPUT 722; "NPLC 0"
```

比／抵抗測定で、最低測定速度での最高確度、最高分解能および80dBのNMRを指定するには、以下を送信します。

```
OUTPUT 722;"NPLC 1000"
```

次のレンジの電源周波数を指定することができます。

- 0～1 PLC (60Hzの基準周波数の場合. 000006 PLCステップで、50Hzの基準周波数の場合. 000005 PLCステップで)
- 1～10 PLC (1 PLCステップで)
- 10～1000 PLC (10 PLCステップで)

注記 10 PLCより大きい積分時間については、10 PLCの積分時間を用いて行なわれた多数の測定の平均値がとられます。例えば、60 PLCを指定した場合、6回の10 PLC測定の平均値がとられます。

広範なPLC設定が可能なため、測定速度、確度、分解能、NMRを自由に選択できます。一般的には、許容できる範囲の分解能とNMRを維持する一方で、十分な速度を実現する積分時間を選択する必要があります。付録Aの仕様一覧に、DC/抵抗測定の場合の積分時間と分解能の桁数およびNMRとの関係が示されています。

積分時間の直接指定

DC/抵抗測定の場合は、APER(アパーチャ)コマンドを使って、積分時間を直接(秒単位で)指定できます。例えば、22 msの積分時間を指定するには、以下を送信します。

```
OUTPUT 722;"APER.022"
```

注記

APERコマンドを使用した場合、本器は、NPLCコマンドを使用した場合のように、長時間にわたる積分時間に対して、読み取り値の平均をとりません。例えば、NPLCコマンドを使ってPLCを60に指定した場合(60 Hzの電源周波数で1秒の積分時間)、マルチメータは6回の10 PLC測定の平均値をとります。APERコマンドを使って1秒の積分時間を指定した場合は、マルチメータは1秒間に1回の読み取りを積分します。

APERコマンドを使えば、500ns〜1sの範囲の積分時間を100ns単位で指定できます。APERコマンドは、信号の特定の部分(パルスなど)をサンプリングする場合や、ディジタイズに最もよく用いられます。APERコマンドを使って、入力信号から特定の周波数の雑音信号を除去することも可能です。このためには、除去する信号の周期の整数倍に等しい積分時間を設定します。例えば、100Hzの周波数(周期 = 10ms)の雑音を除去するには、10ms、20ms、30msといった積分時間を指定します。

分解能の指定

ファンクション・コマンド(FUNC、ACV、DCVなど)またはRANGEコマンド[1] の最後のパラメータとして、測定分解能を指定します。このパラメータは、コマンドの*max._input*パラメータのパーセンテージで指定するため、*%_ resolution*と呼ばれます(本章で前述の「レンジの指定」を参照してください)。本器は、指定した*%_resolution*パラメータに*max._input*パラメータを掛けて、測定分解能を決定します。*%_resolution*パラメータを計算するには、次の式を用います。

%_ resolution = (実際の分解能/最大入力) × 100

例えば、最大予想入力が10 VDCで、必要とされる分解能が1 mVDCであると仮定します。この式では、次のような値が求められます。

%_ resolution = (0.001/10) × 100 = 0.01

*%_resolution*パラメータを省略した場合には、積分時間は、最後に実行されたAPERまたはNPLCコマンドによって指定された時間になります。

1. RESコマンドを使用して、分解能を指定することもできます。使用法を示した例については、第6章の「RESコマンド」を参照してください。

DC/抵抗測定(およびアナログAC測定)については、A/Dコンバータの積分時間によって分解能が決まります。分解能を指定した場合は、実際には、積分時間を間接的に指定していることになります。APERコマンドやNPLCコマンドは積分時間も指定できるので、以下に示すように、分解能を指定した場合には相互作用が生じます。

- 分解能を指定する**前に**APERまたはNPLCコマンドを送信した場合には、本器は、大きい方の分解能(長い方の積分時間)を指定するコマンドを実行します。

- 分解能を指定した**後に**APERまたはNPLCコマンドを送信した場合には、本器は、APERまたはNPLCコマンドによって指定された積分時間を用い、以前に指定された分解能は一切無視されます。

分解能を指定する必要がある場合

DC/抵抗測定については、NPLCまたはAPERコマンドによって指定された分解能が十分でない場合に、分解能を指定する必要があります。例えば、次のプログラムでは、ライン10で、60dBのNMRと7 1/2桁の分解能を提供する1 PLCの積分時間が指定されています。このため、10Vレンジの実際の分解能は1 μVになります。このアプリケーションには、10Vの*max._input*で、100nVの分解能が必要です。前述の式では、0.000001 (1E-6)の*%_resolution*パラメータが得られます。これはライン20に指定されています。

```
10 OUTPUT 722;"NPLC 1"
20 OUTPUT 722;"DCV 10, 1E-6"
30 END
```

オートゼロ

オートゼロ機能は、マルチメータ内部のオフセット誤差が後続のDCまたは抵抗測定から確実にゼロになるようにします。オートゼロ・ファンクションは、AZEROコマンドを使って制御します。AZERO ONでは、マルチメータは入力信号を内部的に切断し、すべての測定に続いてゼロ読み取りを実行します。次に、ゼロ読み取り値を前の測定から代数学的に減算します。AZERO OFFまたはONCEでは、マルチメータはゼロ読み取りを1回実行し、それを後続の読み取り値から減算します。AZERO OFFまたはAZERO ONCEを実行すると、マルチメータは、TARM EXT コマンドが実行されたときにオートゼロ測定を引き起こすTARM EXTを除くすべてのイベントに対して、最初のトリガ・アーム・イベントが発生した時に、オートゼロ測定を実行します。(トリガ・アーム・イベントについては、第4章で説明します)。オートゼロ測定は、測定ファンクション、レンジまたは積分時間が変更される度に更新されます(この種の更新は、トリガ・アーム・イベントが発生するか、TARM EXTが実行された場合に行なわれます)。電源投入時の状態では、AZEROはONに設定されていますが、以下を送信することによって変更することができます。

```
OUTPUT 722;"AZERO OFF"
```

注記

4 端子抵抗測定の場合は、オートゼロをオンのままにしておいてください (AZERO ONコマンド)。オートゼロをオフにしなければならない場合(AZERO OFFまたはONCE)、オートゼロをオフにする前に必ず、測定用の接続をすべて行ない、リード線抵抗が変化しないようにしてください。4端子接続を行なう前にオートゼロをオフにした場合、またはオートゼロがオフの状態ではリード線抵抗が変化する場合(スキャン時など)には、4端子抵抗測定が不正確なものになる恐れがあります。

オフセット補正

抵抗測定には抵抗の電圧の測定が含まれるため、存在するすべての外部起電力(オフセット電圧)が測定確度に影響を及ぼします。オフセット補正がオンになっていると、マルチメータは、オフセット電圧の効果をキャンセルすることによって抵抗測定を補正します。これを実行するには、マルチメータはまず、入力電圧を電流源をオンにした状態で測定します。その後で、電流源はオフにされ、入力電圧が再度測定されます。真の起電力は、2つの測定電圧間の差です。オフセット補正は、2端子と4端子の両方の抵抗測定に対して使用できます。本器で実行できるのは、10Ω～100kΩのレンジでのオフセット補正だけです。オフセット補正は、他のレンジでは機能しません。電源投入時/PRESET NORM状態では、オフセット補正はオフになっています。オフセット補正をオンにするには、以下を送信します。

```
OUTPUT 722:"OCOMP ON"
```

オフセット補正した抵抗測定の最大直列オフセット電圧の仕様については、付録Aを参照してください。

固定入力抵抗

DC電圧測定を実行する場合には、FlXEDZコマンドを使って、マルチメータの入力抵抗を固定することができます。これは、(レンジの変更に起因した)入力抵抗の変動が測定に影響を及ぼさないようにするのに有用です。表12は、FIXEDZ OFFの場合の入力抵抗を示したものです。FIXEDZ ONの場合には、入力抵抗は、すべてのDC電圧レンジに対して一定の10 MΩです。電源投入時/PRESET NORM状態では、固定抵抗はオフ(OFF)になっています。固定抵抗をオンにするには、以下を送信します。

```
OUTPUT 722; "FIXEDZ ON"
```

固定抵抗をオフにするには、以下を送信します。

```
OUTPUT 722;"FIXEDZ OFF"
```

AC測定用の構成

このセクションでは、AC/AC+DC電圧、AC/AC+DC電流、周波数／周期測定を実行するためのマルチメータの構成方法について説明します。

AC/AC+DC電圧

本器は、次の3つの方法のうちの1つを使って、真の実効値AC電圧またはAC+DC電圧測定を実行します: アナログ実効値変換、ランダム・サンプリング変換、同期サンプリング変換。どの測定方法も6つのレンジ(10mV、l00mV、lV、l0V、100V、1000V)を持ち、最高分解能はいずれのレンジでも6 1/2桁です。

表15には、各測定法の測定特性と信号要件が示されています。図11は、あらゆる種類の電圧測定の前面端子の接続を示したものです

AC電圧測定の場合、マルチメータは入力信号のAC成分だけを測定します。AC+DC電圧測定の場合、マルチメータは、表15に示されている周波数レンジ内のDC成分とAC成分を測定します。例えば、アナログ法を使ってAC+DC電圧を測定する場合には、10Hz以下のAC成分は測定には一切含まれないということに注意してください。

注記

何らかのAC測定法を使って10mVと100mVのレンジで測定を実行した場合には、放射ノイズ(大型モータのオン/オフに起因する過渡現象など)によって、読み取り値が不正確になる可能性があります。これらのレンジで正確な測定を行なうためには、電気的に「静寂な」近隣環境を確保し、シールドが施された試験用リード線を使用してください。

表15. ACおよびAC+DC電圧の測定法

ACV/ACDCV 測定法	周波数レンジ	最高確度	繰り返し信号の必要性	1秒当たりの読み取り回数 最低	最高
同期	1 Hz～10MHz	0.01%	必要	0.025	10
アナログ	10 Hz～2MHz	0.03%	不要	0.8	50
ランダム	20 Hz～10MHz	0.10%	不要	0.025	45

同期サンプリング変換

同期サンプリング変換では、サンプルから実効値が計算されますが、繰り返し(周期)入力信号が必要です。同期サンプリングは、直線性に優れており、3つの方法の中で最も正確です。同期サンプリングは、1 Hz～10 MHzの周波数レンジで周期波形を測定する場合に有用です。

同期サンプリングに関する注釈

- 同期サンプリングを行なう場合、マルチメータはLEVEL同期信号源イベント(デフォルト・モード)を使って、サンプリングを入力信号と同期させます。入力信号が読み取り中に除去され、一定の時間内に(制限時間は主に、AC帯域幅の設定(このセクションで後述)によって決まる)返されなかった場合には、測定法がランダム・サンプリングに変わり、測定を実行できるようになります。SSRCコマンドを使って、測定法の変更を防ぐことができます。また、SSRCコマンドを使って、同期サンプリングを**Ext Trig**コネクタの信号に合わせて調整することも可能です。詳細およびサンプル・プログラムについては、第6章の「SSRCコマンド」を参照してください。

- LEVEL同期信号源を使用した場合には、入力信号の雑音によって間違ったレベルのトリガが発生したり、読み取り値が不正確になる可能性があります。正確な測定を行なうためには、電気的に「静寂な」近隣環境を確保し、シールドが施された試験用リード線を使用してください。レベル・フィルタリングをオンにすると(LFILTER ONコマンド)、この種の雑音に対する感度が下がります。詳細については、第6章の「LFILTERコマンド」を参照してください。

- 同期サンプリングの場合は、指定したACVまたはACDCV測定ファンクションに関係なく、入力信号は常にDC結合されます。ACVを指定した場合には、DC成分が読み取り値から数学的に減算されます。AC電圧とDC電圧が結合している場合には、AC電圧だけでは通常生じることのない、過負荷状態が引き起こされる恐れがあるため、これについては十分に考慮することが大切です。

アナログ実効値変換

アナログ実効値変換では、入力信号が直接積分されます。アナログ実効値変換は、電源の投入時に選択される方法です。この方法は、10 Hz〜2 MHzの周波数レンジで信号を測定する場合にうまく働き、3つの方法の中で最も高速の測定を実現します。

ランダム・サンプリング変換

ランダム・サンプリング変換では、生成された読み取り値ごとに、入力信号の多数のサンプルが取り出されます。サンプルには内部タイムベース・ジェネレータによってランダムにスペースが入れられ、信号の真の実効値が統計的に計算されます。ランダム・サンプリングは、同期サンプリングに必要な繰り返し入力信号を必要としないため、広帯域雑音測定などのアプリケーションに最適です。この方法は、直線性に優れている上に、確度が高いため、低レベル(フルスケールの<l/10)の測定に特に適しています。ランダム・サンプリングの測定帯域幅は、20 Hz〜10 MHzです。

AC電圧法の指定

電源を入れると、アナログ実効値変換が選択されます。電源投入時の状態では、次のようにACまたはAC+DC電圧測定を選択するだけで、アナログ実効値変換を使って測定を実行することが可能です。

```
OUTPUT 722;"ACV"                    !AC結合AC電圧測定を選択する
```

または

```
OUTPUT 722; "ACDCV"                 !DC結合AC電圧測定を選択する
```

SETACVコマンドを使って、AC電圧の測定法を指定することができます。例えば、ランダム・サンプリング変換を指定するには、以下を送信します。

```
OUTPUT 722;"SETACV RNDM"
```

同期サンプリング変換を選択するには、以下を送信します。

```
OUTPUT 722;"SETACV SYNC"
```

アナログ実効値変換に戻るには、以下を送信します。

```
OUTPUT 722;"SETACV ANA"
```

指定したAC電圧測定法は、電源を入れ直すか、マルチメータをリセットするか、別の方法を指定するまで有効です。ACまたはAC+DC電圧測定を選択した場合には必ず、最後に指定した(または、電源投入時の)測定法が用いられます。

AC/AC+DC電流

本器は、入力端子間に内部シャント抵抗を入れることによって、その抵抗の電圧を測定し、電流を計算します(電流 = 電圧/抵抗)。ACまたはAC+DC電圧測定とは違って、ACまたはAC+DC電流測定は、アナログ法(直接積分)だけを使って行なうことができます。本器の前面と裏面にある電流入力は、1 A、250Vヒューズによって保護されています。図12は、あらゆる種類の電流測定の前面端子の接続を示したものです。

本器は、5種類のうちのいずれかのレンジでACまたはAC+DC電流を測定します。AC電流測定の場合、マルチメータは入力信号のAC成分だけを測定します。AC+DC電流測定の場合、マルチメータは、DC成分と> 10Hzの周波数レンジ内のAC成分を測定します。AC+DC電流を測定する場合には、10Hz以下のAC成分は測定には一切含まれないということに注意してください。ACまたはAC+DC電流の最高分解能は6

1/2桁です。表16には、各電流レンジとそのフルスケール読み取り値、最高分解能、適用されるシャント抵抗が示されています。(分解能は、指定した積分時間の関数です。詳細については、このセクションで後述の「積分時間の設定」を参照してください)。ACIコマンドを使ってAC電流測定を指定するか、ACDCIコマンドを使ってAC+DC電流測定を指定します。例えば、100µAレンジでのAC電流測定を指定するには、以下を送信します。

```
OUTPUT 722;"ACI 100E-6"
```

10mAレンジでのAC+DC電流測定を指定するには、以下を送信します。

```
OUTPUT 722;"ACDCI 10E-3"
```

表16. ACおよびAC+DC電流レンジと分解能

ACIレンジ	フルスケール読み取り値	最高分解能	シャント抵抗
l00µA	120.0000µA	lOOpA	730Ω
1mA	1.200000mA	1nA	100Ω
10mA	12.00000mA	10nA	10Ω
100mA	120.0000mA	100nA	1Ω
1A	1.050000A	1µA	0.1Ω

周波数または周期

本器の周波数/周期カウンタは、AC電圧またはAC電流入力を許容します。周波数測定と周期測定のいずれの測定の場合も、最高分解能は7桁[1] です。(詳細については、このセクションで後述の「分解能の指定」を参照してください)。

FREQコマンドを使って周波数測定を指定するか、PERコマンドを使って周期測定を指定します。周波数測定または周期測定については、入力信号が電圧源と電流源のいずれからのものであるか、測定がAC結合かDC結合かも指定しなければなりません。これは、FSOURCEコマンドを使って行なわれます(電源投入時/デフォルト値はACVです)。表17は、FSOURCEパラメータ、各パラメータによって指定される入力のタイプ、各パラメータの測定能力を示したものです。電圧源からの周波数または周期測定の端子の接続については、図11を参照してください。電流源からの周波数または周期測定用の端子の接続については、図11を参照してください。

注記 LEVELコマンドは、周波数測定と周期測定については、ゼロ交差しきい値や入力信号結合に影響を及ぼします。詳細については、第6章の「LEVELコマンド」を参照してください。

1. 左端の桁は、ほとんどの測定ファンクションに対しては1/2桁ですが、周波数および周期測定では、フルの桁(0～9)です。

表17. FSOURCEパラメータ

FSOURCE パラメータ	定義	測定能力	
		周波数	周期
ACV	AC結合AC電圧入力	1Hz — 10MHz	100ns — 1s
ACDCV	DC結合AC電圧入力	1Hz — 10MHz	100ns — 1s
ACI	AC結合AC電流入力	1Hz — 100kHz	10µs — 1s
ACDCI	DC結合AC電流入力	1Hz — 100kHz	10µs — 1s

次のプログラムは、電圧源からの10Vレンジでの周波数測定用にマルチメータを構成します。入力信号はAC結合されます。

```
10 OUTPUT 722;"FREQ 10"
20 OUTPUT 722;"FSOURCE ACV"
30 END
```

次のプログラムは、電流源からの10mAレンジでの周期測定用にマルチメータを構成します。入力信号はDC結合されます。

```
10 OUTPUT 722;"PER 10E-3"
20 OUTPUT 722;"FSOURCE ACDCI"
30 END
```

注記 周波数または周期測定については、レベル・フィルタをオンにすることによって、75kHzより上の高周波雑音を減らすことができます。詳細については、第6章の「LFILTERコマンド」を参照してください。

帯域幅の指定

ACBANDコマンドは、あらゆるACおよびAC+DC測定の入力信号の周波数成分を指定します。周波数成分を指定することによって、マルチメータは、正確な測定を行なえるだけでなく、高速測定を行なえるように自己構成することができます。ACBANDコマンドの最初のパラメータは、最小予想周波数成分を指定します(電源投入時/PRESET NORM値は20Hzです)。2番目のパラメータは、最大予想周波数成分を指定します(電源投入時/PRESET NORM値は2MHzです)。例えば、入力信号が750 Hz〜2 kHzの周波数レンジを持つと仮定した場合には、以下を送信します。

```
OUTPUT 722;"ACBAND 750,2000"
```

入力信号の周波数成分に基づいた確度仕様(およびアナログAC測定の測定速度仕様)については、付録A「仕様」を参照してください。

注記 同期ACまたはAC + DC電圧測定の場合は、帯域幅パラメータを使って、タイムアウト値やサンプリング・パラメータが計算されます。オートレンジをオンにした状態での周波数または周期測定の場合は、帯域幅パラメータを使って、オートレンジに必要な期間が決定されます。これらの測定については、指定した帯域幅(特に指定した低周波)が入力信号の周波数成分と一致していることが非常に重要です。

積分時間の設定

積分時間とは、A/Dコンバータが入力信号を測定する期間です。アナログAC測定の場合、積分時間は分解能の最高桁数を決定するだけなく、指定した帯域幅とともに測定速度に影響を及ぼします。(積分時間はまた、アナログAC測定の確度にも少し影響を及ぼします)。アナログAC測定は、アナログ変換法(SETACV ANAコマンド)だけを使って行われるACまたはAC+DC電圧測定であり、ACまたはAC+DC電流測定であると定義されます。積分時間が長い場合には、測定分解能と確度は向上しますが、測定速度は低下します。

注記 積分時間が周波数測定や周期測定に影響を及ぼすことはありません。サンプリングAC電圧測定(SET ACV SYNCまたはSET ACV RNDM)の場合、A/Dコンバータの積分時間は自動的に選択されるため、マルチメータは、サンプリング回数を変更することによって、指定した分解能を実現します(次のセクションの「分解能の指定」を参照)。

アナログAC測定については、NPLCコマンドを使って電源周波数(PLC)に換算して積分時間を指定することができます。(APERコマンドを使って積分時間を指定することも可能です。ただし、これは主にDC測定での使用を目的としています。詳細については、第6章の「APERコマンド」を参照してください)。マルチメータは、指定された数のPLCにA/Dコンバータの基準周波数の周期を乗算して(LFREQコマンド)、積分時間を決定します。例えば、50 Hz電源の周期はl/50 = 20 msになります。10 PLCを指定した場合、積分時間は200 msになります。電源投入時の状態では、積分時間は10 PLCに設定されています。PRESET NORM状態では、積分時間は1 PLCに設定されています。最低確度、4 1/2桁分解能での最高速測定を実現するための積分時間を設定するには、以下を送信します。

```
OUTPUT 722; "NPLC 0"
```

最低測定速度での最高確度および最高分解能(6 1/2桁)を指定するには、以下を送信します。

```
OUTPUT 722; "NPLC 1000"
```

次のレンジの電源周波数を指定することができます。

- 0～1 PLC (60Hzの基準周波数の場合. 000006 PLCステップで、50Hzの基準周波数の場合. 000005 PLCステップで)
- 1～10 PLC (1 PLCステップで)
- 10～1000 PLC (10 PLCステップで)

注記 10 PLCより大きい積分時間については、10 PLCの積分時間を用いて行なわれた多数の測定の平均値がとられます。例えば、60 PLCの積分時間を指定した場合、6回の10 PLC測定の平均値がとられます。

一般的には、許容できる範囲の確度と分解能を維持する一方で、十分な速度を実現する積分時間を選択する必要があります。表18は、アナログAC測定の場合の積分時間と分解能の桁数との関係を示したものです。

表18. アナログAC測定とA/Dコンバータの関係

分解能の桁数	電源周波数(NPLCコマンド)	
	LFREQ = 6OHz	LFREQ = 5OHz
4.5	0 – 0.000030	0 – 0.000025
5.5	0.000036 – 0.000360	0.000030 – 0.000300
6.5	0.000366 – 1000	0.000305 – 1000

分解能の指定

測定分解能は、ファンクション・コマンド(FUNC、ACV、ACIなど)またはRANGEコマンドの最後のパラメータ(*%_resolution*パラメータ)として指定することができます。[1]

すべてのアナログAC電圧および電流測定について、*%_resolution*は、コマンドの*max._input*パラメータのパーセンテージで指定します。本器は、指定した*%_resolution*パラメータに*max._input*パラメータを掛けて、測定分解能を決定します。*%_resolution*パラメータの値を決定するには、次の式を用います。

%_ resolution = (実際の分解能/最大入力) × 100

例えば、最大予想入力が10 VACで、必要な分解能が1 mVACであると仮定します。この式では、次のような値が求められます。

%_ resolution = (0.001/10) × 100 = 0.01

アナログAC測定については、A/Dコンバータの積分時間によって分解能が決まります。分解能を指定した場合は、実際には、積分時間を間接的に指定していることになります。NPLCコマンドは積分時間も指定できるので、以下に示すように、分解能を指定した場合には相互作用が生じます。

- 分解能を指定する**前に**NPLCコマンドを送信した場合には、本器は、大きい方の分解能(長い方の積分時間)を指定するコマンドを実行します。
- 分解能を指定した**後に**NPLCコマンドを送信した場合には、本器は、NPLCコマンドによって指定された積分時間を用い、以前に指定された分解能は無視されます。

1. RESコマンドを使用して、分解能を指定することもできます。使用法を示した例については、第6章の「RESコマンド」を参照してください。

アナログAC測定の場合、*%_resolution*パラメータを省略した場合には、積分時間は、最後に実行されたNPLCコマンドによって指定された時間になります。

サンプリングACVまたはACDCV、ランダム・サンプリング(SETACV RNDM)の分解能は4.5桁の固定分解能で、変更することはできません。同期サンプリング(SETACV SYNC)の場合は、*%_resolution*パラメータが0.001の場合7.5桁、0.01の場合6.5桁、0. 1の場合5.5桁、1の場合4.5桁です。

周波数および周期測定の場合、*%_ resolution*はゲート時間と分解能の桁数を指定します(表19を参照)。例えば、次のプログラムは、10Vレンジを使った、電圧入力からの周波数測定を指定します。ライン20の*%_resolution*パラメータ(.00001)は、1秒のゲート時間と7桁の分解能を指定します。

```
10 OUTPUT 722; "FSOURCE ACV"
20 OUTPUT 722;"FREQ 10,.0000l"
30 END
```

IFREQまたはPER測定の場合、*%_resolution*パラメータを省略した場合には、1秒のゲート時間と7桁の分解能を選択する.0000lに*%_resolution*が設定されます。

表19. 周波数/周期測定のゲート時間と分解能

%_resolution パラメータ	ゲート時間の 選択	分解能の 桁数
.00001	1s	7
.0001	100ms	7
.001	10ms	6
.01	1ms	5
.1	100µs	4

分解能を指定する必要がある場合

アナログACVまたはACDCV (SETACV ANA)、ACI、ACDCIDC測定については、NPLCコマンドによって指定された分解能より高い分解能が必要な場合に、分解能を指定する必要があります。例えば、次のプログラムでは、ライン10で、5 1/2桁の分解能を選択する0.0001 PLCの積分時間が指定されています。このため、実際の分解能は10Vレンジで100 µVになります。ただし、この場合には、10Vの*max._input*で、10 µVの分解能が必要です。前述の式では、0.0001 (1E-4)の*%_resolution*パラメータが得られます。これはライン30に指定されています(この分解能の場合、読み取りには約40秒かかります)。

```
10 OUTPUT 722;"NPLC .0001"
20 OUTPUT 722;"SETACV ANA"
30 OUTPUT 722;"ACV 10,1E-4"
40 END
```

同期サンプリングACVまたはACDCV(SETACV SYNC)、FREQ、PER測定については、分解能の指定が、実際の分解能を変更するための唯一の方法です。これらの測定の場合、積分時間は固定なので、NPLCコマンドと*%_resolution*パラメータとの間に相互作用は生じません。マルチメータは、サンプリング回数を変更することによって、サンプリングした AC 電圧に対して指定した分解能を実現します。(*%_resolution*パラメータを省略した場合には、同期変換法の場合は0.01%に、ランダム変換法の場合は0.4%に*%_resolution*が設定されます)。次のプログラムは、同期サンプリング変換を使って、AC電圧測定を選択します。最大予想入力電圧は10 Vで、

.1の*%_resolution*パラメータは5.5桁の分解能を選択します。このため、実際の分解能は1 mVになります。

```
10 OUTPUT 722; "SETACV SYNC"
20 OUTPUT 722;"ACV 10, .1"
30 END
```

比測定用の構成

比測定の場合、マルチメータは、Ω **Sense**端子に印加されるDC基準電圧と、**Input**端子に印加される信号電圧を測定します。その後で、次のようにしてその比を計算します。

$$比 = \frac{信号電圧}{DC\ 基準電圧}$$

信号電圧の測定ファンクションは、DC電圧、AC電圧またはAC+DC電圧です。(ACまたはAC+DC電圧には、ANA、RNDM、SYNCの3種類の測定法のうちのどれでも使用できます)。基準電圧の測定ファンクションは、常にDC電圧であり、±12VDCの最大測定可能入力を持っています。図15は、比測定用の前面パネルの接続を示したものです。

注記 Ω Sense LO端子とInput LO端子は、共通の基準を持っていなければならず、0.25Vより大きい電圧差を持つことはできません。

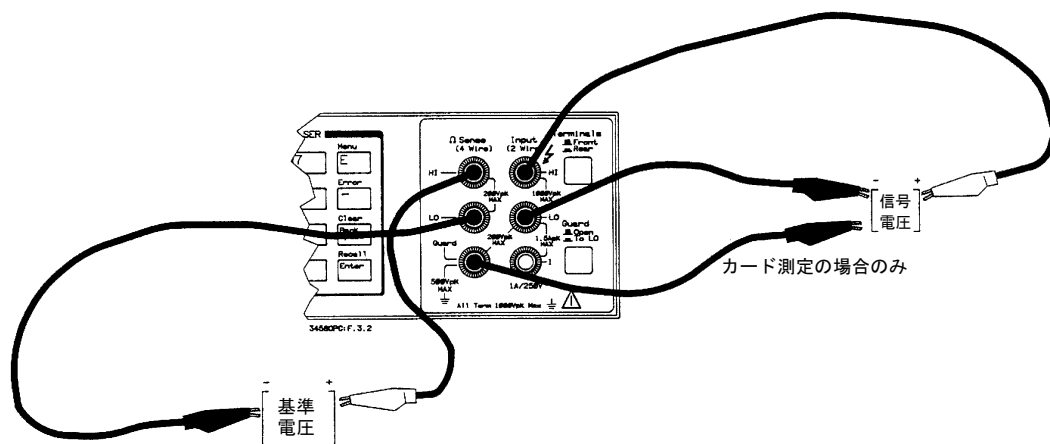



図15. 比測定の接続

比測定の指定

比測定を指定するには、まず信号測定用の測定ファンクション(およびACまたはAC+DC電圧の測定法)を選択し、次にRATIOコマンドを使って比測定をオンにします。例えば、次のプログラムは、同期サンプリング変換を使ったAC電圧比測定(10Vレンジ)を指定します。

```
10 OUTPUT 722;"ACV 10"
20 OUTPUT 722; "SETACV SYNC"
30 OUTPUT 722;"RATIO ON"
40 END
```

その後、比測定をオフにするには、以下を送信します。

```
OUTPUT 722;"RATIO OFF"
```

比測定の場合、指定した測定範囲は、信号電圧測定(**Input**端子)にのみ適用されます。基準電圧測定(Ω **Sense**端子)は、常にオートレンジに設定されます。レンジ切替えについては、本章の最初の方の「一般的な構成」のセクションで詳細に説明しています。

サブプログラム・メモリの使用方法

本器は、コマンド文字列をサブプログラムとして保存することができます。このため、バス/コントローラとの対話を最小限に抑えながら、頻繁に使用されるコマンド文字列を実行することが可能です。保存されたサブプログラムはコンパイルされるため、本器は、GPIBを経由して送信された等価コマンドを実行するよりもはるかに速い速度で、サブプログラムを実行します。本器は、サブプログラムとステート(後述)によって共有される、14kバイトのメモリを備えています。サブプログラム/ステート・メモリが一杯になると、**メモリ・エラー** (エラー・レジスタのビット7)が生成されます。

注記 ステータス・レジスタには、サブプログラムの実行の完了を確認するのに用いることのできる、サブプログラム完了ビットが含まれています。詳細については、本章で後述の「ステータス・レジスタの使用方法」を参照してください。

サブプログラムの保存

サブプログラムは、SUBコマンドやSUBENDコマンドを使って保存します。SUBコマンドは、サブプログラムの開始と識別名を指示します。サブプログラム名には、最大10文字まで含めることができます。また、全部英字の名前か、英数字の組合わせによる名前が可能です(?および_も名前に含めることができます)。英数字名を使用する場合は、最初の文字はアルファベットでなければなりません。英字または英数字のサブプログラム名は、マルチメータのコマンドまたはパラメータ、あるいは保存されているステートの名前と同じであってはいけません。

SUBコマンドに続けて、サブプログラム・コマンドを実行したい順番に入力します。SUBENDコマンドを使って、サブプログラムの終わりを指示します。サブプログラムはすべて、圧縮されていない限り(本章で後述の「サブプログラムの圧縮」を参照)、不揮発性メモリに保存されます(電源を切っても影響を受けません)。例えば、次のプログラムは、ライン20から60までのコマンドを、DCCUR1という名前のサブプログラムとして保存します。

```
10 OUTPUT 722;"SUB DCCUR1"
20 OUTPUT 722;"MEM FIFO"
30 OUTPUT 722;"TRIG HOLD"
40 OUTPUT 722;"DCI 1,.01"
50 OUTPUT 722; "NRDGS 5, AUTO"
60 OUTPUT 722;"TRIG SGL"
70 OUTPUT 722; "SUBEND"
80 END
```

既存のサブプログラムと同じ名前を使って新しいサブプログラムを作成した場合には、新しいサブプログラムによって古いサブプログラムが上書きされます。

サブプログラムの実行

保存されているサブプログラムを実行するには、サブプログラムの名前と一緒にCALLコマンドを発行します。例えば、上述のサブプログラムを実行するには、以下を送信します。

```
OUTPUT 722;"CALL DCCUR1"
```

注記

入力バッファ (本章で後述)がオフの場合は、サブプログラムが完了するか、PAUSEコマンド(以下を参照)が見つかるまで、GPIBは解放されません。サブプログラムの呼出し直後にバスを解放するための方法については、本章で後述の「入力バッファの使用方法」を参照してください。サブプログラムの実行を中断するには、GPIB Device Clearコマンドを送信します。

前面パネルから、CALLコマンドにアクセスし、上または下矢印キーを押すことによって、保存されているすべてのサブプログラムの名前を表示できます。該当するサブプログラムが見つかったら、**Enter**キーを押して、そのサブプログラムを実行します。

サブプログラムの実行の中断

保存されているサブプログラムにPAUSEコマンドを挿入することによって、サブプログラムの実行を一時的に中断できます。本器は、コマンドを1つずつサブプログラムを実行します。PAUSEコマンドに出くわすと、サブプログラムの実行が中断されます。また、そのプログラムがリモートから呼び出されていた場合には、GPIBバスが解放されます。例えば、次のプログラムでは、ライン60にPAUSEコマンドがあります。

```
10 OUTPUT 722; "SUB 2"
20 OUTPUT 722;"MEM FIFO"
30 OUTPUT 722;"TRIG HOLD"
40 OUTPUT 722;"DCV 10"
50 OUTPUT 722;"NRDGS 5,AUTO"
60 OUTPUT 722;"PAUSE"
70 OUTPUT 722;"TRIG SGL"
80 OUTPUT 722;"SUBEND"
90 END
```

上のサブプログラムを呼び出した場合、PAUSEコマンドに至るまでのすべてのコマンドが実行された後で、プログラムの実行が中止します。サブプログラムの実行を再開するには、以下を送信します。

```
OUTPUT 722;"CONT"
```

サブプログラムの実行は、GPIB Group Execute Triggerを送信することによって再開することも可能です(これは、本来読み取りをトリガしません。サブプログラムの動作を再開するに過ぎません)。

ネスト構造のサブプログラム

サブプログラムを使って、別のサブプログラムを呼び出すこと(ネスト構造のサブプログラム)ができます。例えば、次のサブプログラムが呼び出されると(CALL 1コマンド)、10回のDC電圧測定が行なわれた後で、以前に保存されたサブプログラム*DCCUR1*が呼び出されます。

```
10 OUTPUT 722; "SUB 1"
20 OUTPUT 722;"TRIG HOLD"
30 OUTPUT 722;"NRDGS 10,AUTO"
40 OUTPUT 722;"DCV 10"
50 OUTPUT 722;"TRIG SGL"
60 OUTPUT 722; "CALL DCCUR1"
70 OUTPUT 722; "SUBEND"
80 END
```

PAUSEコマンドが含まれているサブプログラムは、別のサブプログラムから呼び出すことはできません。本器では、最大10個までのサブプログラムをネストすることが可能です。つまり、サブプログラム1にサブプログラム2を呼び出させ、サブプログラム2にサブプログラム3を、サブプログラム3にサブプログラム4をというように、サブプログラム10までを呼び出させることができます。

サブプログラムのオートスタート

サブプログラム0という名前を付けた場合、そのサブプログラムは、マルチメータが電源投入時シーケンスを完了するか、前面パネルの**Reset**キーを使ってリセットされる度に実行されます。これは、電源障害が発生した後に、マルチメータを前の状態に自動的に戻す場合に特に有用です。電源障害が検出される度に、マルチメータは、現在の状態をステート0として保存します(ステートについては、本章で後述します)。次のプログラムは、マルチメータを電源切断時状態に戻すオートスタート・プログラムを保存するだけでなく、A/Dコンバータの基準周波数を正確に電源周波数に設定します(詳細については、本章で前述の「基準周波数の変更」を参照してください)。

```
10 OUTPUT 722; "SUB 0"
20 OUTPUT 722;"RSTATE 0"
30 OUTPUT 722;"LFREQ LINE"
40 OUTPUT 722; "SUBEND"
50 END
```

マルチメータの電源を入れ直さずにサブプログラムを実行する必要がある場合にも、オートスタート・サブプログラム(CALL 0コマンド)を呼び出すことが可能です。

サブプログラムの圧縮

サブプログラムを保存する場合、本器は、不揮発性メモリにASCIIテキストを、揮発性メモリにコンパイル済みのサブプログラムをそれぞれ保存します。サブプログラムを呼び出した場合、本器は、コンパイル済みのサブプログラムを実行します(これは、サブプログラムがバスを経由して送られた等価コマンドより速い速度で実行されるためです)。電源を切ると、ASCIIテキストだけが保存されます。電源が入れ直された場合は、本器はASCIIテキストを使って、コンパイル済みのサブプログラムを生成します。サブプログラムは、COMPRESSコマンドを使って圧縮することができます。サブプログラムを圧縮すると、不揮発性メモリからASCIIテキストが削除され、コンパイル済みのものだけが揮発性メモリに残ります。これによって、不揮発性メモリの使用可能なスペースは増えますが、不揮発性メモリからサブプログラムが削除されてしまいます(電源を切るか、前面パネルの**Reset**キーを押すと、サブプログラムの記録がすべて破壊されます)。次のプログラム文は、*DCCUR1*という

名前で以前に保存されたサブプログラムを圧縮します。

```
OUTPUT 722; "COMPRESS DCCUR1"
```

サブプログラムの削除

DELSUBコマンドは、特定のサブプログラムを削除します。例えば、*DCCUR1*という名前のサブプログラムを削除するには、以下を送信します。

```
OUTPUT 722; "DELSUB DCCUR1"
```

SCRATCHコマンドを使って、保存されているすべてのプログラムとステートをメモリから削除することも可能です。

ステート・メモリの使用方法

マルチメータの現在の構成(測定ファンクション、レンジ、分解能、積分時間など)を、特定のステートとしてステート・メモリに保存することができます。サブプログラム、読み取り値および一部の演算レジスタの内容(詳細については、第6章の「SSTATEコマンド」を参照)は、保存されたステートには含まれていません。電源の遮断が生じた場合には、本器は現在の構成をステート0に保存します。このため、ステートを位置0に保存した場合には、電源の切断時に、位置0に保存したステートが現在の構成によって上書きされてしまいます。本器は、サブプログラムとステートの両方に用いられる14kバイトのメモリを備えています。各ステートが約300バイトを占めます。メモリ内にサブプログラムが1つもない場合には、最高46個までのステートを保存できます。サブプログラム/ステート・メモリが一杯になると、**メモリ・エラー** (エラー・レジスタのビット7)が生成されます。

ステートの保存

SSTATEコマンドは、マルチメータの現在の状態を識別名とともに保存します。ステート名には、最大10文字まで含めることができます。また、全部英字の名前か、英数字の組合わせによる名前が可能です(?および_も名前に含めることができます)。また、0～127の範囲の整数を名前として使用することも可能です(これは主に前面パネル操作用です)。英数字名を使用する場合は、最初の文字はアルファベットでなければなりません。英字または英数字のステート名は、マルチメータのコマンドまたはパラメータ、あるいは保存されているサブプログラムの名前と同じであってはなりません。整数のステート名を使用した場合は、ステートの保存時に、プレフィックス*STATE*が整数に割り当てられます。これによって、整数のステート名と整数のサブプログラム名が区別されます。例えば、名前8で保存されるステートは、*STATE8*として記録されます。このステートは、8と*STATE8*のいずれの名前ででも、リコールすることができます。

ステートはすべて、不揮発性メモリに保存されます(電源を切っても影響を受けません)。本器は、保存されると同時にステートをコンパイルします。このため、ステートがリコールされると、マルチメータは、ステートを作成するのに用いられた個々のコマンドを実行する方法よりもはるかに速い速度で自己構成します。現在のマルチメータの状態を*ACST1*という名前のステートとして保存するには、以下を送信します。

```
OUTPUT 722; "SSTATE ACST1"
```

ステートのリコール

RSTATEコマンドは、メモリからステートをリコールし、リコールしたステートに合わせてマルチメータを構成します。例えば、ステートACSTlをリコールするには、以下を送信します。

```
OUTPUT 722;"RSTATE ACST1"
```

前面パネルから、RSTATEコマンドにアクセスし、上または下矢印キーを押すことによって、保存されているすべてのステートの名前を表示できます。該当するステートが見つかったら、**Enter**キーを押して、ステートをリコールします。

ステートの削除

PURGEコマンドを用いることによって､保存されている単一のステートを削除することができます。例えば、ステート*ACST1*を削除するには、以下を送信します。

```
OUTPUT 722;"PURGE ACST1"
```

SCRATCHコマンドを使って、保存されているすべてのステートとサブプログラムをメモリから削除することも可能です。

入力バッファの使用方法

マルチメータの電源投入時/PRESET NORM状態では､入力バッファはオフになっています。このため、マルチメータは、各GPIBコマンドを個別に処理し、コマンドが実行されるのを待ってからGPIBバスを解放するか､別のコマンドを受け入れなければなりません。ほとんどの場合、コントローラは、バスが解放されるのを待ってから処理を続行して、コントローラと測定器間の同期を確保しなければなりません。これは、実行に長時間を要するコマンドでは最も顕著です。例えば、リモートからセルフテストを実行した場合(TESTコマンド)､セルフテストが完了するまで50秒近く待たなければ、GPIBバスは解放されません。

入力バッファがオンになっている場合には、マルチメータは、コマンドをバッファに一時的に保存して、GPIBバスをすぐに解放します。その後で、マルチメータは、入力バッファからコマンドを1つ1つ受け取った順番に実行します。このため、コントローラは、マルチメータがコマンドを実行している間も、他の処理を実行することができます。次のプログラムは、TESTコマンドを実行する前に、入力バッファをオンにします。

```
10 OUTPUT 722;"INBUF ON"
20 OUTPUT 722; "TEST"
30 END
```

入力バッファは、最高255文字まで保持できます。入力バッファの許容量を超える文字を送信した場合は、バッファ・スペースが使用可能になるまで、バスはホールド状態になります。スペースが使用可能になると、残りの文字が入力バッファに入れられ、バスが解放されます。

入力バッファを使用している場合は、バッファ内のすべてのコマンドが実行された時のことを認識して置く必要があるかも知れません。本器は、ステータス・レジスタ(次を参照)にビット4(命令準備完了)を設定して、この情報を提供します。ステータス・レジスタが適切にオンになっていないと、GPIBのSRQ(サービス・リクエスト)ラインが真になります｡SRQを割り込みとして受け入れるように予めプログラムされている場合には、コントローラはこれに対して確認応答します。

ステータス・レジスタの使用方法

ステータス・レジスタは、次のマルチメータのステータス情報をモニタします。

- サブプログラムの完了
- 上限値/下限値の超過
- SRQコマンド実行
- 電源投入
- 命令準備完了
- エラー
- サービス・リクエスト
- データ使用可能

これらの事象のうちの1つが発生すると、対応するビットがステータス・レジスタに設定されます。以下に、ステータス・レジスタの各ビットの意味の定義をリストします。

ビット0(重み = 1) サブプログラムの完了－保存されているサブプログラムの実行が完了しました。

ビット1(重み = 2) 上限値/下限値の超過－1つまたは複数の読み取り値が合否判定演算の指定された上限値/下限値を超過しました。このビットは、リアルタイム演算と後処理演算の両方に適用されます。(第4章の「合否判定」を参照してください)。

ビット2(重み = 4) SRQコマンド実行－マルチメータのSRQコマンドが実行されました。

ビット3(重み = 8) 電源投入－電源投入時シーケンスが発生しました。

ビット4(重み = 16) 命令準備完了－マルチメータは、以前のコマンドの実行をすべて完了したため、さらにコマンドを受け入れることができます。(TRIG SGLまたはTARM SGLを使って入力バッファをオフにした状態で一連の読み取りを開始した場合は、このビットを使ってすべての読み取りの完了をモニタすることができます)。

ビット5(重み = 32) エラー－1つまたは複数のエラーがエラー /補助レジスタに記録されました。詳細については、本章で前述の「エラー・レジスタの 読み取り」を参照してください。

注記

EMASK コマンドを用いることによって、何らかのエラーまたはすべてのエラーによって、ステータス・レジスタにエラー・ビットが設定されるのを防ぐことができます。詳細については、第6章の「EMASKコマンド」を参照してください。

ビット6(重み = 64) サービス・リクエスト－サービスが要求され、GPIB SRQラインが真に設定されています。このビットは、ステータス・レジスタのどれか別のビットが設定され、RQSコマンドによってSRQをアサートできるようにされた場合に設定されます。ビット6は、エラーによってエラー・レジスタにビットが設定され、それによって今度はビット6が設定される場合などに設定される唯一のビットとなることができます。後で、エラー・レジスタは読み取られます。エラー・ビットは削除されますが、ビット6は設定されたままになっています。

ビット7(重み = 128) データ使用可能－出力バッファの読み取りまたは問合せ応答が使用可能です。

ステータス・レジスタの読み取り

STB?問合せコマンドは、ステータス・レジスタを読み取り、すべての設定ビットの重み和を返します。STB?コマンドは、ステータス・レジスタをクリアしません。次のプログラムは、STB?コマンドを使って、ステータス・レジスタの内容を読み取ります。

```
10 OUTPUT 722."STB?"
20 ENTER 722; A
30 PRINT A
40 END
```

例えば、ビット3(重み = 8)とビット7(重み = 128)が設定されていると仮定します。上のプログラムは、2つの重みの和(136)を返します。

マルチメータはSTB?コマンドの処理で忙しいので、レディ状態にないため、STB?コマンドによりビット4(命令準備完了)が設定されているかわかりません。レディ・ビットをモニタしたい場合は、GPIB Serial Pollコマンドを使って、ステータス・レジスタを読み取らなければなりません。SRQラインが真の場合は、Serial Pollコマンドはすべてのステータス・レジスタ・ビットをクリアします。* ビット6がクリアされた場合は、SRQラインは偽に戻されます。Serial Pollコマンドの実行中にSRQラインが偽である場合には、レジスタの内容は変更されません。次のプログラムは、Serial Pollコマンドを使ったステータス・レジスタの読み取り方法を示しています。

```
10 P=SPOLL(722)
20 DISP P
30 END
```

ステータス・レジスタ[1] をクリアするには、以下を送信します。

```
OUTPUT 722; "CSB"
```

割り込み

ステータス・レジスタの中の1つのビットが設定され、SRQをアサートできるようになっている場合には(RQSコマンド)、GPIB SRQラインが真に設定されます。これを使って、現在の処理を中断して、マルチメータが要求するサービスを見つけ出すように、コントローラに警告することができます。割り込みに応答するようにプログラムするための方法については、コントローラの操作マニュアルを参照してください)。

ステータス・レジスタのいずれかのビットがSRQラインを真に設定できるようにするには、RQSコマンドを使って該当するビットをまずオンにする必要があります。例えば、上限値または下限値を超えた場合(ビット1)、電源が入れ直された場合(ビット3)、またはエラーが発生した場合(ビット5)に、アプリケーションが割り込みを要求すると仮定します。これらのビットの等価10進値は、それぞれ2、8、32になります。10進値の和は42になります。以下を送信することによって、これらのビットがSRQをアサートできるようにすることができます。

```
OUTPUT 722;"RQS 42"
```

1. ビットを設定する条件がまだ存在する場合は、ビット4、5、および6はクリアされません。

これで、ビット1、3または5に関連するイベントが発生した場合には必ず、ビット6がステータス・レジスタに設定され、SRQがアサートされます。オンになっていないビットは依然として対応する条件に応答することに注意してください。ただし、それらのビットがビット6を設定したり、SRQをアサートすることはありません。次のプログラムは、HP シリーズ200/300 BASICを使った割り込みの一例です。

```
10 ！上限値 / 下限値超過、エラー、電源の入れ直しによる割り込み
20 OUTPUT 722;"PRESET NORM"
30 OUTPUT 722; "CSB"
40 ON INTR 7 GOTO 90
50 ENABLE INTR 7;2
60 OUTPUT 722;"RQS 42;MATH PFAIL;SMATH MIN -5;SMATH MAX
5"
70 OUTPUT 722;"TRIG AUTO"
80 GOTO 80
90 OUTPUT 722; "STB?"
100 ENTER 722;A
110 IF BINAND (A,2) THEN PRINT "HI/LO LIMIT EXCEEDED"
120 IF BINAND (A,8) THEN PRINT "POWER WAS CYCLED"
130 IF BINAND (A,32) THEN PRINT "ERROR OCCURRED"
140 END
```

ライン20は、マルチメータを初期設定します。これによって、トリガが中断されます。ライン30は、ステータス・レジスタをクリアします。ライン40は、割り込みが発生した場合はライン90に進むように、コントローラに指示します。ライン50は、GPIBインタフェース上でのSRQ割り込みをオンにします。ライン60は、上限値/下限値ビット、パワーオン・ビットおよびエラー・ビットがSRQをアサートできるようにします。ライン60も、下限値–5、上限値+5でのリアルタイム合否判定演算をオンにします。ライン70は、自動トリガをオンにします。ライン80は、コントローラが割り込みを待つようにします。ライン90〜130は、ステータス・レジスタを読み取り、割り込みを引き起こした条件を出力します。

第4章 測定の実行

第4章　測定の実行

概要

本章では、測定のトリガ方法、読み取り値フォーマット、読み取り値メモリの使用方法、GPIBバスを経由した読み取り値の転送方法について説明します。さらに、読み取り速度やGPIBバス転送速度を上げたり、読み取り速度を確認するための方法、本マルチメータのEXTOUT信号や演算機能の使用方法についても説明します。

測定のトリガ

本器が読み取りを行なう前に、3つの個別のイベントが適切な順番で発生する必要があります。これらのイベントとは、(1)トリガ・アーム・イベント、(2)トリガ・イベント、(3)サンプリング・イベントです。サブ・サンプリング(第5章を参照)と複数のトリガ・アーミング(本章で説明)は、このトリガ階層の唯一の例外です。図16に示されているように、3つのイベントすべてがリストされている順番で発生した場合に、マルチメータは指定された読み取りを開始します。電源投入時の状態では、マルチメータは読み取りを自動的に行なうように構成されています。つまり、3つのイベントすべてがAUTOに設定されています。ほとんどのアプリケーションでは、これらのイベントのうちの1つまたは2つだけを使い、残りのイベントはAUTOに設定したままにしておく必要があります。このセクションでは、トリガ・アーム、トリガおよびサンプリングのすべてのイベントの要件を満たすために用いることのできる各種イベントについて説明するとともに、それらのイベントの使用方法を例を挙げて説明します。

注記　本書の中で取り上げられている例では、BASIC言語を使用するHewlett-Packardシリーズ200/300コンピュータを対象としています。これらの例では、GPIBインタフェース・セレクト・コードが7、機器アドレスが22、それらを組み合わせたGPIBアドレスが722と仮定されています。このセクションの例の中には、読み取り値をメモリに保存するものもあれば、読み取り値をコントローラに転送するものもあります。読み取り値の宛先については、本章の「読み取り値メモリの使用方法」および「バスによる読み取り値の送信」の中で詳細に説明します。

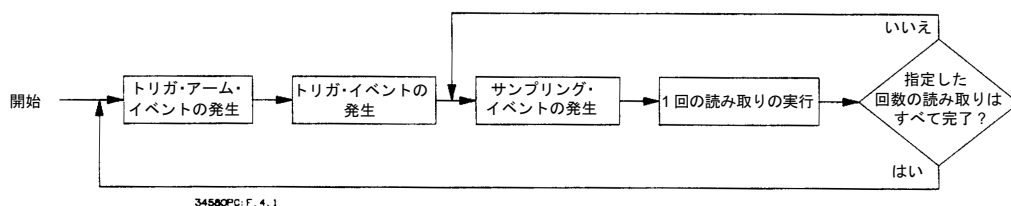



図16. トリガ階層

トリガ・アーム・イベント

指定したトリガ・アーム・イベントが発生すると、マルチメータのトリガ・メカニズムがアーミングされます。すなわち、トリガ・アーム・イベントは、後続のトリガ・イベントを可能にします。トリガ・アーム・イベントは、TARMコマンドを使って指定します。

トリガ・イベント

指定したトリガ・イベントが発生すると(トリガ・アーム・イベントがすでに発生していると)、後続のサンプリング・イベントがオンになります。トリガ・イベントは、TRIGコマンドを使って指定します。

サンプリング・イベント

サンプリング・イベントが発生すると(トリガ・アーム・イベントとトリガ・イベントがすでに発生していると)、マルチメータは読み取りを実行します。マルチメータは、指定した回数の読み取りが行なわれるまで、サンプリング・イベント当たり1回の読み取りを実行します。NRDGSコマンドの最初のパラメータ(読み取り回数)は、トリガ・イベント当たりの読み取り回数を指定します。2番目のパラメータは、各読み取りを開始するイベント(サンプリング・イベント)を指定します。

イベントの選択項目

多種多様なイベントの中から選択して、トリガ・アーム・イベント、トリガ・イベント、サンプリング・イベントとして使用することができます。表20には、イベント・パラメータの説明と、イベント・パラメータが適用されるコマンドが示されています。

表20. イベント・パラメータ

イベント・パラメータ	TARMで使用	TRIGで使用	NRDGS	イベントの説明
AUTO	•	•	•	自動的に発生(必要な場合には必ず)
EXT	•	•	•	マルチメータの外部トリガ入力の立下がりエッジで発生
HOLD	•	•		測定を中断
LEVEL [1]		•	•	指定した電圧が入力信号の指定したスロープに達した場合に発生
LINE [2]		•	•	電源電圧が0Vと交差した場合に発生
SGL	•	•		1回発生(TARM SGLまたはTRIG SGLコマンドの受信時に発生、その後HOLDになる)
SYN	•	•	•	マルチメータの出力バッファが空で、読み取り値メモリがオフまたは空で、コントローラがデータを要求した場合に発生
TIMER [2]			•	一定の読み取り間隔で自動的に発生

1 LEVELトリガまたはサンプリング・イベントは、DC電圧または直接サンプリングしたディジタイズにのみ使用可能です。

2 TIMERまたはLINEイベントは、同期法やランダム法を使ったAC/AC+DC電圧測定、または周波数/周期測定には使用できません。

連続読み取りの実行

電源投入時の状態では、マルチメータのトリガ・アーム、トリガおよびサンプリングのすべてのイベントがAUTOに設定されています。このため、読み取りが連続して行なわれます。一般的には、連続読み取りは、TARM HOLDかTRIG HOLDのいずれかのコマンドを使って、またはマルチメータをPRESET状態のうちの1つに設定することによってマルチメータを構成する前に、中断する必要があります(第3章の「読み取りの停止」を参照)。マルチメータの構成後に、以下を送信することによって、連続読み取りを再開することが可能です(他のトリガ・イベントは変更されていないものと仮定します)。

```
OUTPUT 722;"TARM AUTO”
!TARM HOLD、PRESET FASTまたはPRESET DIGによって、中断された読み取りを再開します
```
または
```
OUTPUT 722; "TRIG AUTO"
!TRIG HOLDまたはPRESET NORMによって読み取りを再開します
```

1回の読み取りの実行

NRDGSコマンドは、各読み取りを開始するトリガ・イベントおよびサンプリング・イベント当たりの読み取り回数を指定します。電源投入時には、RESET、PRESET NORMまたはPRESET FAST状態、トリガ当たりの読み取り回数は1に設定されている他、サンプリング・イベントはAUTOになっています(NRDGS 1、AUTO)。これらのいずれの状態でも、読み取りを中断しているイベントがあれば、そのイベントに応じてTARM SGLコマンドかTRIG SGLコマンドを実行することによって、1回の読み取りを開始することができます。例えば、次のプログラムは、マルチメータをリセットした後で、トリガ・アーム・イベントをHOLDに設定することによって読み取りを中断します。構成が変更され(ライン30〜50)、ライン60で1回の読み取りが開始され、それがコントローラに転送され、表示されます。1回の読み取りの後に、トリガ・アーム・イベントがHOLDになり、読み取りが中断されます。

```
10 OUTPUT 722;"RESET"                    !リセット、全トリガ・イベントAUTO
20 OUTPUT 722;"TARM HOLD"                !読み取り中断
30 OUTPUT 722;"DCV 10"                   !DC電圧、10Vレンジ
40 OUTPUT 722;"NPLC 1"                   !1 PLC積分時間
50 OUTPUT 722;"AZERO OFF"                !オートゼロ・オフ
60 OUTPUT 722;"TARM SGL"                 !1回の読み取りをトリガ
70 ENTER 722;A                           !読み取り値を入力
80 PRINT A                               !読み取り値を出力
90 END
```

PRESET NORM状態では、トリガ・イベントがSYNに設定されるため、読み取りは中断されます(SYNイベントについては、本章で後述します)。この状態では、TRIG SGLコマンドを使って、1回の読み取りを開始することができます。例えば、次のプログラムでは、ライン10で、トリガ・イベントをSYNに設定することによって読み取りが中断されます。ライン20は1回の読み取りを開始し、その読み取り値がコントローラに転送され、表示されます。TRIG SGLコマンドの実行に続いて、トリガ・イベントがHOLDになり、読み取りが中断されます。

```
10 OUTPUT 722;"PRESET NORM"              !TARM AUTO、TRIG SYN、NRDGS 1、AUTO
20 OUTPUT 722;"TRIG SGL"                 !シングル・トリガを生成
30 ENTER 722;A                           !読み取り値を入力
40 PRINT A                               !読み取り値を出力
50 END
```

複数の読み取りの実行

NRDGSコマンドを使って、トリガ・イベント当たり2回以上の読み取りを指定することができます。例えば、次のプログラムは、トリガ・イベント当たり10回の読み取りを実行し(サンプリング・イベントの場合は1イベント当たり1個の値が読み取とられます)、読み取り値がコントローラに転送されます。入力バッファはオンになっていることに注意してください(ライン40)。これは、入力バッファがオフになっていると、SGLイベント(ライン60)は、指定したすべての読み取りが完了するまで、GPIBバスをホールド状態にするからです。これによって、ライン70は、最後の読み取り値を除くすべての読み取り値をコントローラに転送しなくなります。入力バッファをオンにすると、TRIG SGLコマンドはバスをホールド状態になるのを防ぎ、使用可能になると、各読み取り値が転送されます。

```
10 OPTION BASE 1                     !配列の番号付けは1から開始
20 DIM Rdgs(lO)                      !10個の読み取り値用の配列を宣言
30 OUTPUT 722;"PRESET NORM"          !TARM AUTO、TRIG SYN、DCV AUTORANGE
```

```
40 OUTPUT 722;"INBUF ON"              !入力バッファ・オン
50 OUTPUT 722;"NRDGS 10, AUTO"        !10回の読み取り/トリガ、自動サンプリング・イベント
60 OUTPUT 722;"TRIG SGL"              !読み取りをトリガ
70 ENTER 722;Rdgs(*)                  !読み取り値を入力
80 PRINT Rdgs(*)                      !読み取り値を表示
90 END
```

複数のトリガ・アーミング

TARMコマンドの2番目のパラメータは、ユーザが複数のトリガ・アーミングを指定できるようにします。複数のトリガ・アーミングを指定した場合、トリガ・アーム・イベントの1回の発生で、マルチメータが指定した回数だけアーミングされます。(複数のアーミングの場合は、トリガ・アーム・イベントはSGLでなければなりません)。これによって、マルチメータは複数のグループの読み取りを実行します(図17を参照)。

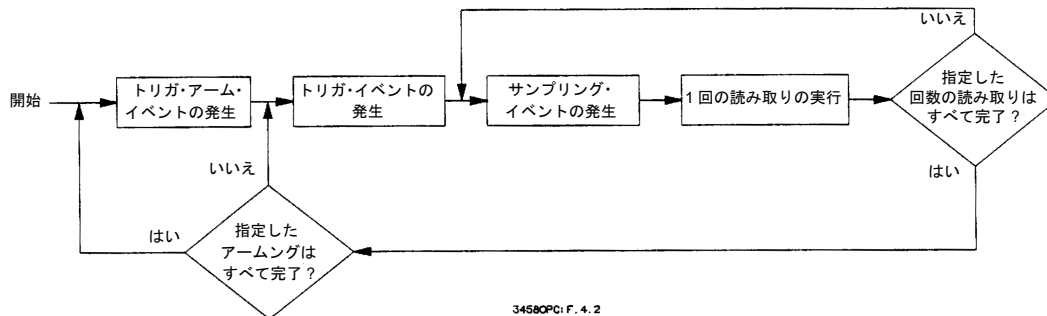



図17. 複数のトリガ・アーミング

次のプログラムでは、NRDGSコマンドによって、トリガ・イベント当たり10回の読み取りが選択されています。TARMコマンドの2番目のパラメータは、5回のアーミングを指定します。このプログラムは、全部で50回の読み取りに対して、10個の読み取り値を5グループ保存します。

```
10  OPTION BASE 1                  !配列の番号付けは1から開始
20  DIM Rdgs(50)                   !50個の読み取り値用の配列を宣言
30  OUTPUT 722;"PRESET NORM"       !TARM AUTO、TRIG SYN、DCV AUTORANGE
40  OUTPUT 722;"TARM HOLD"         !トリガ・アーム・イベントをホールド
50  OUTPUT 722;"TRIG AUTO"         !自動トリガ・イベント
60  OUTPUT 722;"INBUF ON"          !入力バッファをオン
70  OUTPUT 722;"NRDGS 10,AUTO"     !10回の読み取り/トリガ、自動サンプリング・イベント
80  OUTPUT 722;"TARM SGL,5"        !アーム・トリガ5回
90  ENTER 722;Rdgs(*)              !読み取り値を入力
100 PRINT Rdgs(*)                  !読み取り値を出力
110 END
```

同期読み取りの実行

トリガ・アーム・イベント、トリガ・イベントまたはサンプリング・イベントを同期(SYN)に設定することによって、マルチメータをコントローラと同期させることができます。同期イベントは、マルチメータの出力バッファが空で、読み取り値メモリがオフか空で、コントローラがデータを要求した場合に必ず発生します。つまり、コントローラが測定を必要とする度に、測定が実行されます。これは、特にマルチメータが高速モードの場合、リモート操作に欠かせない非常に重要な機能の1つです。

高速モードでは、同期イベントによって、コントローラがいつでも読み取り値を受け入れられるようになるため、読み取り速度が低下しません。詳細については、本章で後述の「高速モード」を参照してください。次のプログラムでは、PRESET

NORMコマンドによって、トリガ・イベントが同期に設定されています。ライン40は、同期トリガ・イベント当たり15回の読み取りを指定します。ライン50は、マルチメータからのデータを要求します。これによって、同期トリガ・イベントの条件が満たされ、読み取りが開始されます。ライン50はマルチメータからのデータを15回要求していることに注意してください。複数の読み取りが指定され、SYNがトリガ・イベントまたはトリガ・アーム・イベントとして用いられている場合には、マルチメータは複数のデータ・リクエストを個別のSYNイベントとして認識しません。つまり、このプログラムでは、SYNトリガ・イベントは、15回ではなく、1回発生します。

```
10 OPTION BASE 1                    !配列の番号付けは1から開始
20 DIM Rdgs (15)                    !15個の読み取り値用の配列を宣言
30 OUTPUT 722;"PRESET NORM"         !TARM AUTO、TRIG SYN、DCV AUTORANGE、MEM OFF
40 OUTPUT 722;"NRDGS 15,AUTO"       !15回の読み取り/トリガ、自動サンプリング・イベント
50 ENTER 722;Rdgs(*)                !SYNイベントを生成、読み取り値を入力
60 PRINT Rdgs(*)                    !読み取り値を出力
70 END
```

次のプログラムでは、同期イベントをサンプリング・イベントとして使用しています。ライン60は、マルチメータからのデータを15回要求します。SYNがサンプリング・イベントとして使用されている場合は、各データ・リクエストは1つのSYNイベントとして認識されます。すなわち、このプログラムでは、SYNイベントが15回発生しています。

```
10 OPTION BASE 1                    !配列の番号付けは1から開始
20 DIM Rdgs(15)                     !15個の読み取り値用の配列を宣言
30 OUTPUT 722;"PRESET NORM"         !TARM AUTO、TRIG SYN、DCV AUTORANGE
40 OUTPUT 722;"NRDGS 15,SYN"        !15回の読み取り/トリガ、同期サンプリング
50 OUTPUT 722;"TRIG AUTO"           !自動トリガ・イベント
60 ENTER 722;Rdgs(*)                !SYNイベント、各読み取り値を入力
70 DISP Rdgs(*)                     !読み取り値を出力
80 END
```

タイマ読み取りの実行

トリガ当たり複数の読み取りを実行する場合、TIMERサンプリング・イベントを使って、読み取り間に指定された時間間隔を置くことができます。この間隔は、1回の読み取りの開始から次の読み取りの開始までの期間です。TIMERコマンドを使って、秒単位で間隔を指定します。(指定の間隔が各読み取りを実行するのに必要な時間に満たない場合には、マルチメータはTRlG TOO FASTエラーを生成します)。次のプログラムは、1秒間隔での読み取りで、トリガ当たり8回の読み取りを指定します(図18を参照)。

```
10 OPTION BASE 1                    !配列の番号付けは1から開始
20 DIM Rdgs(8)                      !8個の読み取り値用の配列を宣言
30 OUTPUT 722;"PRESET NORM"         !TARM AUTO、TRIG SYN、DCV AUTORANGE
40 OUTPUT 722;"NRDGS 8, TIMER"      !8回の読み取り/トリガ、タイマ・サンプリング・イベント
50 OUTPUT 722;"TIMER 1"             !1秒のタイマ間隔
60 ENTER 722;Rdgs(*)                !SYNイベント、各読み取り値を入力
70 PRINT Rdgs(*)                    !読み取り値を出力
80 END
```

SWEEPコマンドを使って、NRDGS n,TIMERコマンドとTlMERコマンドを交換することも可能です。SWEEPコマンドの最初のパラメータは読み取り間の間隔を指定し、2番目のパラメータは読み取り回数を指定します。(SWEEPコマンドとNRDGSコマンドは交換可能です。マルチメータはプログラミングで最後に指定された方を使用します)。例えば、次のプログラムは、1秒間隔での読み取りで、トリガ当たり8回の読み取りを指定します(図18を参照)。

```
10 OPTION BASE 1                       !配列の番号付けは1から開始
20 DIM Rdgs(8)                         !8個の読み取り値用の配列を宣言
30 OUTPUT 722;"PRESET NORM"            !TARM AUTO、TRIG SYN、DCV AUTORANGE
40 OUTPUT 722;"SWEEP 1,8"              !1秒間隔、8回の読み取り/トリガ
50 ENTER 722; Rdgs(*)                  !SYNイベント、各読み取り値を入力
60 PRINT Rdgs(*)                       !読み取り値を出力
70 END
```

注記　TIMERサンプリング・イベントまたはSWEEPコマンドを使用している場合は、オートレンジはオフになります。TIMERまたはSWEEPは、同期法やランダム法(SETACV SYNCまたはRNDM)を使ったAC/AC+DC電圧測定、または周波数/周期測定には使用できません。

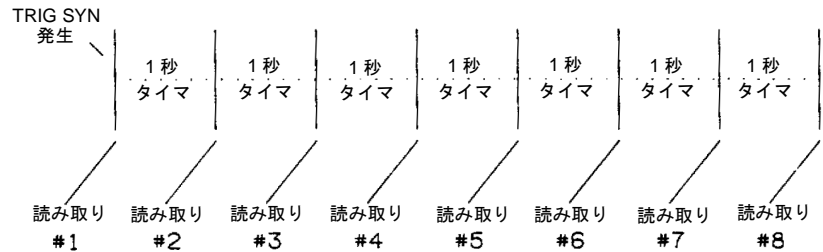



図18. タイマまたは掃引間隔

遅延読み取りの実行

DELAYコマンドを用いることによって、トリガ・イベントと最初のサンプリング・イベントとの間に挿入される時間間隔を指定することができます。例えば、次のプログラムでは、指定した遅延間隔は2秒、掃引間隔は1秒です。ライン40は、トリガ・イベント当たり8回の読み取りを指定します。図19からは、トリガ・イベント(TRIG SGL)と最初の読み取りとの間で、遅延が発生していることが分かります。次に、続く読み取り間に、掃引間隔が発生しています。この例では、測定全体に追加される時間は9秒です。

```
10 OPTION BASE 1                 !配列の番号付けは1から開始
20 DIM Rdgs(8)                   !読み取り値用の配列を宣言
30 OUTPUT 722;"PRESET NORM"      !TARM AUTO、TRIG SYN、DCV AUTORANGE
40 OUTPUT 722;"SWEEP 1,8"        !1秒間隔、8回の読み取り/トリガ
50 OUTPUT 722;"DELAY 2"          !2秒の遅延
60 ENTER 722;Rdgs(*)             !読み取り値を入力
70 PRINT Rdgs(*)                 !読み取り値を出力
80 END
```

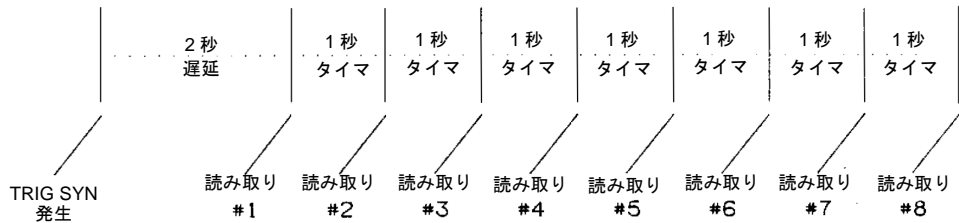



図19. 掃引(またはタイマ)による遅延

デフォルト遅延

遅延間隔を指定しなかった場合は、現在の測定ファンクション、レンジ、分解能およびAC帯域幅設定に基づいて、マルチメータが自動的に遅延時間を決定します(デフォルト遅延時間)。この遅延時間は、実際には、読み取り前の許容セトリング時間であり、これにより正確な測定が実現されます。デフォルト遅延時間は、ファンクションのレンジ、分解能、またはAC帯域幅が変更される度に、自動的に更新されます。ただし、一度遅延時間値を指定すれば、RESETまたはPRESETコマンドを実行するか、電源を入れ直すか、別の遅延値を指定するか、遅延パラメータをデフォルト値に戻すまで(自動遅延に戻るDELAY -1コマンド)、指定した遅延時間値は変わりません。次のプログラムでは、DELAY?問合せコマンドを使って、PRESET NORM状態の遅延時間を応答します。

```
10 OUTPUT 722;"PRESET NORM"
20 OUTPUT 722;"DELAY?"
30 ENTER 722;A$
40 PRINT A$
50 END
```

外部トリガ

外部(EXT)イベントによって、マルチメータは外部信号源からトリガされるようになります。このイベントは、トリガ・アーム・イベント、トリガ・イベントまたはサンプリング・イベントとして使用できます。EXTイベントは、マルチメータの裏面パネルにある**Ext Trig**コネクタに送られるTTLパルスの立下がりエッジで発生します。認識できる最小パルス幅は250nsです。外部トリガ回路の帯域幅は5MHzです。

次のプログラムでは、EXTイベントをトリガ・イベントとして用いています。サンプリング・イベントはAUTOです。トリガ・イベント当たりの読み取り回数は1に設定されています。**Ext Trig**端子の立下がりエッジに達すると、マルチメータは読み取りを実行し、それがコントローラに転送されます。2番目の立下がりエッジによって2番目の読み取りが開始され、それがコントローラに転送されます。このシーケンスは、20回すべての読み取りが完了し、コントローラに転送されるまで続きます。

```
10 OPTION BASE 1                    !配列の番号付けは1から開始
20 DIM Rdgs(20)                     !読み取り値用の配列を宣言
30 OUTPUT 722;"PRESET NORM"         !TARM AUTO、TRIG SYN、NRDGS l、AUTO、
50 OUTPUT 722;"TRIG EXT"            !各読み取りをトリガ
60 ENTER 722;Rdgs(*)                !読み取り値を入力
70 PRINT Rdgs(*)                    !読み取り値を出力
80 END
```

次の例では、EXTをサンプリング・イベントとして用いています。トリガ・イベントは同期です(PRESET NORMコマンドによって選択)。トリガ・イベント当たりの読み取り回数は10に設定されています。コントローラがライン50を実行すると、同期イベントが発生し、サンプリング・イベント(EXT)がオンになります。**Ext Trig**端子に立下がりエッジが達すると、マルチメータは1回の読み取りを実行し、それがコントローラに転送されます。2番目の立下がりエッジによって2番目の読み取りが開始され、それがコントローラに転送されます。このシーケンスは、10回すべての読み取りが完了し、コントローラに転送されるまで続きます。

```
10 OPTION BASE 1                    !配列の番号付けは1から開始
20 DIM Rdgs(l0)                     !読み取り値用の配列を宣言
30 OUTPUT 722;"PRESET NORM"         !TARM AUTO、TRIG SYN、DCV AUTORANGE
40 OUTPUT 722;"NRDGS 10,EXT"        !10回の読み取り/トリガ、外部サンプリング・イベント
50 ENTER 722;Rdgs(*)                !読み取り値を入力
60 PRINT Rdgs(*)                    !読み取り値を出力
70 END
```

注記 マルチメータを外部スキャナと同期させるための方法を示した例については、本章で後述の「EXTOUT信号」を参照してください。

外部トリガのバッファリング

トリガのバッファリングは、外部(EXT)トリガ・アーム、トリガまたはサンプリングのいずれかのイベントを使用している場合に発生する可能性のあるエラー(TRIGGER TOO FAST)に対する補正を行ないます。トリガのバッファリングをオフにした場合には、読み取り中に発生した外部トリガ信号はいずれもTRIGGER TOO FASTエラーを生成するため、トリガは無視されます。トリガのバッファリングをオンにした場合には、読み取り中に発生した最初の外部トリガが保存されるため、このトリガまたは続くトリガによってエラーが生成されることはありません。読み取りの完了後には、保存されたトリガはEXTイベントの条件を満たします。ただしこれは、マルチメータがそのようにプログラムされている場合です。トリガのバッファリングは、入力終了(ICOMP)イベントを使ってマルチメータのEXTOUT信号と同期がとられた外部スキャナを使用している場合に有用です。ICOMPパルスは各読み取りが完了する前に発生するため、スキャナは次のチャネルを閉じて、読み取りが完了する前に、チャネル・クローズ・パルス(マルチメータにトリガをかけるのに使用)を生成します。(詳細については、本章で後述の「入力完了」を参照してください)。マルチメータの電源投入時の状態では、トリガのバッファリングはオフになっています。トリガのバッファリングをオンにするには、以下を送信します。

```
OUTPUT 722;"TBUFF ON"
```

トリガのバッファリングをオフにするには、以下を送信します。

```
OUTPUT 722; "TBUFF OFF"
```

イベントの組合わせ

アプリケーションに合わせて、多数のトリガ・アーム・イベント、トリガ・イベントおよびサンプリング・イベントの組合わせを指定することができます。表21には、これらのイベントの可能な組合わせがすべて示されている他、その結果としての各々のトリガ・シーケンスについて説明されています。

表21. イベントの組合わせ

トリガ・アーム イベント	トリガ イベント	サンプリング イベント	説明
AUTO	AUTO	任意	サンプリング・イベント当たり1個の値が読み取られる(サンプリング・イベントがAUTOの場合は、複数の読み取りが連続して行なわれる)
AUTO	EXT	AUTO, EXT, TIMER, LINE, LEVEL	**Ext Trig**入力で立下がりエッジが生じた後に、指定した回数の読み取りが完了するまで、サンプリング・イベント当たり1個の値が読み取られる
AUTO	EXT	SYN	不正
AUTO	LEVEL	AUTO, EXT, TIMER, LEVEL	LEVELイベントが発生すると[1]、指定した回数の読み取りが完了するまで、サンプリング・イベント当たり1個の値が読み取られる
AUTO	LEVEL	SYN, LINE	不正
AUTO	LINE	AUTO, EXT, TIMER, LINE	電源電圧が0Vと交差すると、指定した回数の読み取りが完了するまで、サンプリング・イベント当たり1個の値が読み取られる
AUTO	LINE	SYN, LEVEL	不正
AUTO	SGL	任意	TRIG SGLコマンドを実行すると、指定した回数の読み取りが完了するまで、サンプリング・イベント当たり1個の値が読み取られる。トリガ・イベントがHOLDになる。SYNサンプリング・イベントを使用する場合は、入力バッファがオンになっているか、TRIG SGLコマンドの送信時にcr lfを抑制する必要がある
AUTO	SYN	SYN	コントローラがデータを要求すると[2]、両方のSYNイベントの条件が満たされ、最初の読み取りが行なわれる。指定した回数の読み取りが完了するまで、SYNイベント当たり1個の値が読み取られる
AUTO	SYN	AUTO, EXT, LEVEL, LINE, TIMER	コントローラがデータを要求すると[2]、指定した回数の読み取りが完了するまで、サンプリング・イベント当たり1個の値が読み取られる
EXT	AUTO	Any	**Ext Trig**入力で立下がりエッジが生じた後に、指定した回数の読み取りが完了するまで、サンプリング・イベント当たり1個の値が読み取られる
EXT	EXT	AUTO, EXT, TIMER, LINE, LEVEL	**Ext Trig**入力で2つの立下がりエッジが生じた後に、指定した回数の読み取りが完了するまで、サンプリング・イベント当たり1個の値が読み取られる
EXT	EXT	SYN	不正
EXT	LEVEL	AUTO, EXT, TIMER, LEVEL	**Ext Trig**入力で立下がりエッジが生じた後にLEVELイベントが発生すると[1]、指定した回数の読み取りが完了するまで、サンプリング・イベント当たり1個の値が読み取られる
EXT	LEVEL	SYN, LINE	不正

[1] LEVELイベントは、指定した電圧が入力信号の指定したスロープに達した場合に発生します。LEVELトリガ・イベントまたはサンプリング・イベントは、DC電圧または直接サンプリング測定にのみ使用可能です。

[2] SYNイベントが発生するためには、出力バッファが空で、読み取り値メモリがOFFまたは空でなければなりません。

[3] 入力バッファがオンになっていなければなりません。オンの場合は、TARM SGLコマンドの送信時にcr lfを抑制する必要があります。

表21. イベントの組合わせ

トリガ・アーム イベント	トリガ イベント	サンプリング イベント	説明
EXT	LINE	AUTO, EXT, TIMER, LINE	**Ext Trig**入力で立下りエッジが生じた後に電源電圧のゼロ交差が発生すると、指定した回数の読み取りが完了するまで、サンプリング・イベント当たり1個の値が読み取られる
EXT	LINE	SYN, LEVEL	不正
EXT	SGL	任意	不正
EXT	SYN	SYN	**Ext Trig**入力で立下りエッジが生じた後にコントローラがデータ(両方のSYNイベントの条件を満たす)を要求すると[2]、最初の読み取りが行なわれる。さらに、指定した回数の読み取りが完了するまで、SYNイベント当たり1個の値が読み取られる
EXT	SYN	AUTO, EXT, TIMER, LINE, LEVEL	**Ext Trig**入力で立下りエッジが生じた後にコントローラがデータを要求すると[2]、指定した回数の読み取りが完了するまで、サンプリング・イベント当たり1個の値が読み取られる
HOLD	任意	任意	トリガ・アーム・イベントが変更されるまで、読み取りは行なわれない
AUTO, EXT, SGL, SYN	HOLD	任意	トリガ・イベントが変更されるまで、読み取りは行なわれない。SGLトリガ・アーム・イベントとSYNサンプリング・イベントを使用する場合は、入力バッファがオンになっているか、TARM SGLコマンドの送信時にcr lfを抑制する必要がある
SGL	AUTO	任意	TARM SGLコマンドを実行すると、指定した回数の読み取りが完了するまで、サンプリング・イベント当たり1個の値が読み取られる。トリガ・アーム・イベントがHOLDになる。SYNサンプリング・イベントを使用する場合は、入力バッファがオンになっているか、TARM SGLコマンドの送信時にcr lfを抑制する必要がある
SGL	EXT	AUTO, EXT, TIMER, LINE, LEVEL	TARM SGLコマンドの実行後に**Ext Trig**入力で立下がりエッジが生じると、指定した回数の読み取りが完了するまで、サンプリング・イベント当たり1個の値が読み取られる。トリガ・アーム・イベントがHOLDになる
SGL	EXT	SYN	不正
SGL	LEVEL	AUTO, EXT, TIMER, LEVEL	TARM SGLコマンドの実行後にLEVELイベントが発生すると[1]、指定した回数の読み取りが完了するまで、サンプリング・イベント当たり1個の値が読み取られる。トリガ・アーム・イベントがHOLDになる
SGL	LEVEL	SYN, LINE	不正
SGL	LINE	AUTO, EXT, TIMER, LINE	TARM SGLコマンドの実行後に電源電圧のゼロ交差が発生すると、指定した回数の読み取りが完了するまで、サンプリング・イベント当たり1個の値が読み取られる。トリガ・アーム・イベントがHOLDになる
SGL	LINE	SYN, LEVEL	不正
SGL	SGL	任意	不正

[1] LEVELイベントは、指定した電圧が入力信号の指定したスロープに達した場合に発生します。LEVELトリガ・イベントまたはサンプリング・イベントは、DC電圧または直接サンプリング測定にのみ使用可能です。

[2] SYNイベントが発生するためには、出力バッファが空で、読み取り値メモリがOFFまたは空でなければなりません。

[3] 入力バッファがオンになっていなければなりません。オンの場合は、TARM SGLコマンドの送信時にcr lfを抑制する必要があります。

表21. イベントの組合わせ

トリガ・アーム イベント	トリガ イベント	サンプリング イベント	説明
SGL	SYN	SYN	TARM SGLコマンドの実行後にコントローラが両方のSYNイベントの条件を満たすデータを要求すると[2]、最初の読み取りが行なわれる。さらに、指定した回数の読み取りが完了するまで、SYNイベント当たり1個の値が読み取られる。[3] トリガ・アーム・イベントがHOLDになる
SGL	SYN	AUTO, EXT, TIMER, LINE, LEVEL	TARM SGLコマンドの実行後にコントローラがデータを要求すると[2]、指定した回数の読み取りが完了するまで、サンプリング・イベント当たり1個の値が読み取られる。[3] トリガ・アーム・イベントがHOLDになる
SYN	AUTO	SYN	コントローラがデータを要求すると[2] (両方のSYNイベントの条件を満たす)、最初の読み取りが行なわれる。さらに、指定した回数の読み取りが完了するまで、SYNイベント当たり1個の値が読み取られる
SYN	AUTO	AUTO, EXT, TIMER, LINE, LEVEL	コントローラがデータを要求すると[2]、指定した回数の読み取りが完了するまで、サンプリング・イベント当たり1個の値が読み取られる。
SYN	EXT	AUTO, EXT, TIMER, LINE, LEVEL	コントローラがデータを要求した後に[2] **Ext Trig**入力で立下がりエッジが生じると、指定した回数の読み取りが完了するまで、サンプリング・イベント当たり1個の値が読み取られる
SYN	EXT	SYN	不正
SYN	LEVEL	AUTO, EXT, TIMER, LEVEL	コントローラがデータを要求した後に[2] LEVELイベントが発生すると[1]、指定した回数の読み取りが完了するまで、サンプリング・イベント当たり1個の値が読み取られる
SYN	LEVEL	SYN, LINE	不正
SYN	LINE	AUTO, EXT, TIMER, LINE	コントローラがデータを要求した後に[2]電源電圧のゼロ交差が発生すると、指定した回数の読み取りが完了するまで、サンプリング・イベント当たり1個の値が読み取られる
SYN	LINE	SYN, LEVEL	不正
SYN	SGL	任意	不正
SYN	SYN	SYN	コントローラがデータを要求すると[2]、3つのイベントすべての条件が満たされ、最初の読み取りが行なわれる。指定した回数の読み取りが完了するまで、SYNイベント当たり1個の値が読み取られる
SYN	SYN	AUTO, EXT, TIMER, LINE, LEVEL	コントローラがデータを要求すると[2]、両方のSYNイベントの条件が満たされる。指定した回数の読み取りが完了するまで、サンプリング・イベント当たり1個の値が読み取られる

[1] LEVELイベントは、指定した電圧が入力信号の指定したスロープに達した場合に発生します。LEVELトリガ・イベントまたはサンプリング・イベントは、DC電圧または直接サンプリング測定にのみ使用可能です。

[2] SYNイベントが発生するためには、出力バッファが空で、読み取り値メモリがOFFまたは空でなければなりません。

[3] 入力バッファがオンになっていなければなりません。オンの場合は、TARM SGLコマンドの送信時にcr lfを抑制する必要があります。

読み取り値のフォーマット

このセクションでは、読み取り値の保存またはGPIBへの読み取り値の出力に使用することのできるASCII、単精度整数(SINT)、倍精度整数(DINT)、単精度実数(SREAL)および倍精度実数(DREAL)の各フォーマットについて説明します。メモリへの読み取り値の保存については、本章の「読み取り値メモリの使用方法」の中で後述します。GPIBへの読み取り値の出力については、本章の「バスによる読み取り値の送信」の中で後述します。

ASCII

ASCIIフォーマットは、読み取り値当たり15バイトで、V、A、Ω、Hz、またはs(以下を参照)の標準単位で、科学的記数法を用いてコード化されています。

SD.DDDDDDDDESDD

ここで、
S = 記号(+または–)
D = 0〜9
E = 仮数部と基数10の指数部の間の区切り記号

単精度整数と倍精度整数

単精度(SINT)フォーマットは読み取り値当たり2バイトで、倍精度整数(DINT)フォーマットは読み取り値当たり4バイトでそれぞれ構成されます。いずれのフォーマットでも、2の補数コーディングが用いられます。

注記

SINTまたはDINTメモリ/出力フォーマットを使用した場合、マルチメータはスケール・ファクタを読み取り値に適用します。スケール・ファクタは、マルチメータの測定ファンクション、レンジ、A/Dコンバータの設定、オンになっている演算に基づきます。SINTまたはDINTフォーマットは、周波数/周期測定(リアルタイム/後処理演算がオンになっている場合(STATまたはPFAILを除く)、またはオートレンジがオンになっている場合)には使用しないでください。

2の補数バイナリ・コーディング

2の補数バイナリ・コーディングは、2進数によって正の整数と負の整数の両方を表現できるようにする方法です。2の補数コーディングは、記号(実際には、最上位ビット(MSB)の等価10進数)を変更することによって行なわれます。MSBが(1)に設定された場合、1バイトの2の補数では、値は1×– (2^7) = –128になります。MSBが(0)にリセットされた場合、値は0×– (2^7) = 0になります。8ビット、1バイトの2の補数の範囲は、0〜255ではなく、–128〜127であることに注意してください。

次の例では、この2の補数ワードの等価10進数を求めます。

10110101 10010110

この2の補数ワードは以下と等価です。

$-(2^{15}) + 2^{13} + 2^{12} + 2^{10} + 2^8 + 2^7 + 2^4 + 2^2 + 2^1$

次の値が求められます: –19050

単精度実数

単精度実数(SREAL)フォーマットは、IEEE-754仕様に準拠しています。このフォーマットは、次のように、読み取り値当たり32ビット、4バイトから構成されます。

```
S EEE EEEE  E MMM MMMM  MMMM MMMM  MMMM MMMM
  byte 0       byte 1     byte 2    byte 3
```

ここで:

S = 記号ビット(1 = 負 0 = 正)
E = 127でバイアスされた基数2の指数部(これら8ビットを「デコード」するには、等価10進数から127を減算します)。
M = 仮数部ビット(小数点の右側部分のビット)。小数点の左側には、暗黙の最上位ビット(MSB)があります。このビットは、常に、「1」であると仮定されます。これによって、重み2^{-23}の最下位ビット(右端)を伴った、24ビットの有効精度が提供されます。この仮数部の数値を求めるためのもう1つの方法としては、これら24ビット(MSBを「1」と仮定)を整数に変換した後で、2^{-23}を乗算するという方法があります。

SREALフォーマットの数の値は、次の式によって計算されます。

$(-1)^s \times (\text{仮数部}) \times 2^{(\text{指数部})}$

SREALの例

この例では、次のSREALフォーマットの数の等価10進数を求めます。

```
SEEEEEEE EMMMMMMM MMMMMMMM MMMMMMMM
10111011 11001000 01001000 10010000
```

記号ビット「S」は「1」に設定されます。これは、負の数であることを表わします。

基数2の指数部(01110111)の値は次のように求められます。

$2^6 + 2^5 + 2^4 + 2^2 + 2^1 + 2^0 = 119$

この指数は127でバイアスされているため、実数値は次のようになります。

指数部 – 127 = 119 – 127 = –8

仮数部 [1.10010000100100010010000 (MSBを「1」と仮定)] の値は次のように求められます。

$1 + 2^{-1} + 2^{-4} + 2^{-9} + 2^{-12} + 2^{-16} + 2^{-19} = 1.56471443177$

次のように、仮数部をビット・レベルではなく、バイト・レベルで評価します。

```
  バイト1    バイト2    バイト3  =  バイト1  バイト2    バイト3
11001000 01001000 10010000   200     72       144
```

仮数部 $=200 \times 2^{-7} + 72 \times 2^{-15} + 144 \times 2^{-23} = 1.56471443177$

または

仮数部 $= (200 \times 2^{16} + 72 \times 2^8 + 144) \times 2^{-23} = 1.56471443177$

SREAL数は、次の式によって計算されます。

$-1 \times 2^{-8} \times 1.56471443177 = -6.1121657491E-3$

倍精度実数 倍精度実数(DREAL)フォーマットは、IEEE-754仕様に準拠しており、次のように読み取り値当たり64ビット(8ビット)から構成されます。

```
  バイト0      バイト1     バイト2     バイト3
S EEE EEEE  EEEE MMMM  MMMM MMMM  MMMM MMMM
  バイト4      バイト5     バイト 6    バイト7
MMMM MMMM   MMMM MMMM  MMMM MMMM  MMMM MMMM
```

ここで、

S = 記号ビット(1 = 負 0 = 正)
E = 1023によってバイアスされた基数2の指数部(これら11ビットを「デコード」するには、等価10進数から1023を減算します)。
M = 仮数部ビット(小数点の右側部分のビット)。小数点の左側には、暗黙の最上位ビット(MSB)があります。このビットは、常に「1」です。これによって、重み2^{-52}の最下位ビット(右端)を伴った、53ビットの有効精度が提供されます。この仮数部の数値を求めるためのもう1つの方法としては、これら53ビット(MSB =「1」)を変換する方法があります(整数に変換した後で、2^{-52}を乗算します)。

DREALフォーマットの数の値は、次の式によって計算されます。

$(-1)^{S} \times (\text{仮数部}) \times 2^{(\text{指数部})}$

読み取り値メモリの使用方法

本器は、読み取りの実行中に、読み取り値メモリがオンである場合には必ず、読み取り値をメモリに保存します。読み取り値メモリには、FIFO (first-in-first-out)モードとLIFO (last-in-first-out)モードがあります。FIFOモードでは、読み取り番号を指定せずに読み取り値をリコールした場合には(本章で後述の暗黙の読み取り方法)、最初に保存された読み取り値が最初に返されます。FIFOモードで読み取り値メモリをすべて埋めた場合には、保存されているすべての読み取り値はそのまま残り、新しい読み取り値は保存されません。

LIFOモードでは、読み取り番号を指定せずに読み取り値をリコールした場合には、最後に保存された読み取り値が最初に返されます。LIFOモードにある読み取り値メモリをすべて埋めた場合には、最も古い読み取り値が最も新しい読み取り値に置き換えられます。読み取り値メモリをオンにし、MEMコマンドを使ってモードを指定します。(メモリ読み取りモードを指定すると、前に保存した読み取り値がすべて消去されます)。例えば、LIFOモードを使ったメモリ読み取りを指定するには、以下を送信します。

```
OUTPUT 722."MEM LIFO"
```

マルチメータは、現在、読み取り値を保存できる状態になっています。読み取り値の保存後は、以下を送信することによって、読み取り値メモリをオフにし、保存されているすべての読み取り値をそのままの状態にしておくことができます。

```
OUTPUT 722;"MEM OFF"
```

後で、以下を送信することによって、前のモードを再開して、保存されているすべての読み取り値をクリアすることなく、追加の読み取り値を保存することができます。

```
OUTPUT 722;"MEM CONT"
```

メモリ・フォーマット

読み取り値は､次の5種類のフォーマットのうちの1つを使って保存できます: ASCII､単精度整数(SINT)、倍精度整数(DINT)、単精度実数(SREAL)または倍精度実数(DREAL)。各フォーマットに必要なメモリ空間は以下の通りです。

ASCII -	16バイト/読み取り値[1]
SINT -	2バイト/読み取り値
DINT -	4バイト/読み取り値
SREAL -	4バイト/読み取り値
DREAL -	8バイト/読み取り値

特定のフォーマットを使って保存できる読み取り値の個数を確認するには、読み取り値メモリ・サイズ(MSIZE?コマンドによって返される最初の応答)を、上に示されている読み取り値当たりのバイト数で割ります。

- **単精度整数(SINT)または倍精度整数(DINT)** 固定レンジ(オートレンジをオフ)での高速・低分解能測定(3.5桁または4.5桁)には、SINTメモリ・フォーマットを使用します。(SINTフォーマットは読み取り値当たり2バイトに過ぎないので、この他のいずれのメモリ・フォーマットよりも、SINTを使った方がより多くの読み取り値を保存できます)。固定レンジでの高速・高分解能測定(5.5.桁以上)には、DINTメモリ・フォーマットを使用します。

注記 SINTまたはDINTメモリ・フォーマットを使用した場合、マルチメータはスケール・ファクタを読み取り値に適用します。スケール・ファクタは、マルチメータの構成(測定ファンクション、レンジ、A/Dコンバータの設定、オンになっている演算)に基づきます。読み取り値をリコールした場合、マルチメータは現在の構成に基づいてスケール・ファクタを計算します。読み取り値が保存されてから以降に構成が変更された場合には、別のスケール・ファクタが用いられ、読み取り値に誤りが生じる可能性があります。保存されている読み取り値をリコールする場合は、マルチメータを読み取り値が保存された時と同じ構成であることが非常に重要です。SINTまたはDINTフォーマットは、周波数/周期測定(リアルタイム/後処理演算がオンになっている場合(STATまたはPFAILを除く)、またはオートレンジがオンになっている場合)には使用しないでください。

- **単精度実数(SREAL)または倍精度実数(DREAL)** SINTおよびDINTフォーマットとは違い、SREALまたはDREALフォーマットで保存されている読み取り値は、スケーリングされていないので､どのような測定ファンクション/マルチメータ構成ででも使用できます｡(スケール･ファクタはないので､SREALおよびDREALフォーマットは、オートレンジ/演算機能がオンになっている場合に適しています)。6.5桁以下の分解能での測定には、SREALフォーマットを使用します。6.5桁より大きな分解能での測定には、DREALフォーマットを使用します。

1. ASCIIフォーマットは、実際には、読み取り値用の15バイトに加えて、保存されたASCIIの読み取り値の分割にのみ使用されるヌル文字用に、1つの読み取り値当たり1バイトがあります。

- ASCII このメモリ・フォーマットは、どのような測定ファンクション/マルチメータ構成にも使用できます。ASCIIは読み取り値当たりのバイト数が最大なので、出力フォーマットがASCIIで、読み取り速度が重要でなく、保存される読み取り値の個数が多くない場合にだけ使用してください。

MFORMATコマンドは、読み取り値メモリのフォーマットを指定します(電源投入時のデフォルト・フォーマットはSREALです)。例えば、単精度整数フォーマットを選択するには、以下を送信します。

```
OUTPUT 722; "MFORMAT SINT"
```

過負荷表示

本器は、読み取る代わりに、値±1E+38を読み取り値メモリに保存することによって、過負荷状態を知らせます(現在のレンジを上回る入力を測定できます)。過負荷値がディスプレイにリコールされた場合、値±1E+38が表示されます。過負荷値が読み取り値メモリからGPIB出力バッファに転送された場合､値は指定の出力フォーマットの過負荷番号に変換されます。詳細については、本章で後述の「バスによる読み取り値の送信」を参照してください。

読み取り値のリコール

読み取り番号または「暗黙の読み取り」と呼ばれる方法を使って、読み取り値メモリから読み取り値をリコールすることができます。リコールされた読み取り値は、指定した読み取り値メモリ・フォーマットに関係なく、OFORMATコマンドで指定したフォーマットで出力されます(詳細については、本章で後述の「バスによる読み取り値の送信」を参照してください)。読み取り値をリコールする前に、保存されている読み取り値の個数を確認したい場合があるかも知れません。これは、MCOUNT?問合せコマンドを使って行なうことができます。次のプログラムは、保存されている読み取り値の総数を返します。

```
10 OUTPUT 722;"MCOUNT?"
20 ENTER 722;A
30 PRINT A
40 END
```

読み取り番号の使用方法

本器は、読み取り値メモリ内の1つ1つの読み取り値に番号を割り当てます。最新の読み取り値には最小番号(1)が､最も古い読み取り値には最大の番号がそれぞれ割り当てられます。LIFOとFIFOのいずれのモードが使用されているかに関わらず、読み取り番号は常にこの方法で割り当てられます。RMEMコマンドを用いることによって、読み取り番号を使って、1個の読み取り値または読み取り値のグループを、メモリから出力バッファにコピーすることが可能です。RMEMコマンドがメモリ内の読み取り値を破壊することはありません。このコマンドは、読み取り値を出力バッファにコピーするだけです。

RMEMコマンドは、読み取り値メモリをオフにします。つまり､以前に保存された読み取り値はすべてそのまま残り､新しい読み取り値は保存されません。RMEMコマンドの最初のパラメータは、最初の読み取り(最初のパラメータ)を指定します。2番目のパラメータ(count)は、最初の読み取りから始まるリコールされる読み取り値の個数を指定します。3番目のパラメータ(record)は、読み取り値のリコール元のレコードを指定します。レコードは、NRDGSまたはSWEEPコマンドで指定された読み取り値の個数に対応します。例えば、NRDGSコマンドで4個の読み取り値を指定した場合は、読み取り値メモリ内の各レコードには、4個の読み取り値が含まれます｡次のプログラムは､トリガ当たり6個の読み取り値を指定し(NRDGS 10)､TARM

SGLコマンドを使って10個の読み取り値を8回読み取ります(複数のトリガ・アーミング)。これによって、全部で80個の読み取り値がメモリに入れられます。

```
10 OUTPUT 722;"TARM HOLD"          !読み取りを中断
20 OUTPUT 722."DCV 1"              !DC電圧、1Vレンジ
30 OUTPUT 722;"MEM FIFO"           !読み取り値メモリをオン、FIFOモード
40 OUTPUT 722;"TRIG AUTO"          !自動トリガ・イベント
50 OUTPUT 722;"NRDGS 10,AUTO"      !10回の読み取り/トリガ、自動サンプリング・イベント
60 OUTPUT 722;"TARM SGL,8"         !アーム・トリガを8回
70 END
```

これで、保存されている読み取り値へは、個々の読み取り番号(1〜80)またはレコード/読み取り番号で(例えば、レコード2の3番目の読み取り値は読み取り番号13です)アクセスできます。例えば、次のプログラムは、読み取り番号50(上のプログラムによって読み取られる31番目の読み取り値)を返し、表示します。

```
10 OUTPUT 722;"RMEM 50"          !読み取り番号50をリコール
20 ENTER 722;A                   !読み取り値を入力
30 PRINT A                       !読み取り値を出力
40 END
```

次のプログラムは、最初のパラメータとcountパラメータを使って、12〜17までの番号の読み取り値を返し、表示します

```
10 OPTION BASE 1                 !配列の番号付けは1から開始
20 DIM Rdgs(6)                   !6個の読み取り値用の配列を宣言
30 OUTPUT 722;"RMEM 12,6"        !6個の読み取り値をリコール、#12から開始
40 ENTER 722;Rdgs(*)             !読み取り値を入力
50 PRINT Rdgs(*)                 !読み取り値を出力
60 END
```

読み取り値をリコールする場合に、レコード番号を用いることも可能です。本器は、最小のレコード番号(1)を最新レコードに、最大のレコード番号を最も古いレコードにそれぞれ割り当てます。次のプログラムは、レコード番号6の3番目と4番目の読み取り値を返します(この場合は、それぞれ53と54という読み取り番号が付けられています)。

```
10 OPTION BASE 1                 !配列の番号付けは1から開始
20 DIM Rdgs(2)                   !読み取り値用の配列を宣言
30 OUTPUT 722;"RMEM 3,2,6"       !レコード#6から3番目と4番目の読み取り値をリコール
40 ENTER 722;Rdgs(*)             !読み取り値を入力
50 PRINT Rdgs(*)                 !読み取り値を出力
60 END
```

前面パネルからRMEMを実行している場合は、読み取り番号で読み取り値をリコールした後に、上または下矢印キーを使ってメモリ内の他の読み取り値をスクロールすることができます。(RMEMコマンドは、保存されている読み取り値を前面パネルから検索するための唯一の方法です)。

暗黙の読み取りの使用方法

コントローラがマルチメータのデータを要求した場合、出力バッファが空で、読み取り値メモリがオンになっている場合には、読み取り値がメモリから削除され、出力バッファに入れられ、さらにコントローラに転送されます。これが、読み取り値をリコールするための「暗黙の読み取り」法です。RMEMコマンドと違って、暗黙の読み取り法では、メモリから読み取り値が削除されます。LIFOモードでは、最新の読み取り値が返されます。FIFOモードでは、最も古い読み取り値が返されます。次のプログラムは、200回の読み取りを実行し、読み取り値を読み取り値メモリに入れ、暗黙の読み取り法を使って読み取り値をコントローラに転送します。

```
10 OPTION BASE 1                         !配列の番号付けは1から開始
20 DIM Rdgs(200)                         !200個の読み取り値用の配列を宣言
30 OUTPUT 722;"PRESET NORM"              !TARM AUTO、TRIG SYN、DCV AUTORANGE
40 OUTPUT 722;"NRDGS 200,AUTO"           !200回の読み取り/トリガ、自動サンプリング・イベント
50 OUTPUT 722;"MEM FIFO"                 !読み取り値メモリをオン、FIFOモード
60 OUTPUT 722;"TRIG SGL"                 !読み取りをトリガ
70 PAUSE                                 !プログラムを一時停止、再開するにはCONTINUEを押す
80 ENTER 722;Rdgs(*)                     !読み取り値を入力
90 PRINT Rdgs(*)                         !読み取り値を出力
100 END
```

バスによる読み取り値の送信

このセクションでは、読み取り値の出力フォーマットと、マルチメータからコントローラへの読み取り値の転送方法について説明します。

出力フォーマット

本器は、読み取りを実行している時に読み取り値メモリがオフである場合(MEM OFFコマンド)には必ず、読み取り値をGPIB出力バッファに送信します。(電源投入時、RESETまたはPRESET状態のうちのいずれかの状態にある場合には、読み取り値メモリはオフになっています)。5種類の出力フォーマットと読み取り値当たりのバイト数を以下に示します。

```
ASCII     --   15  バイト/読み取り値
SINT      --    2  バイト/読み取り値
DINT      --    4  バイト/読み取り値
SREAL     --    4  バイト/読み取り値
DREAL     --    8  バイト/読み取り値
```

- ASCII スケール・ファクタを持たず、コントローラによるデータの変換に特殊な処理を必要としないため、最もよく用いられている出力フォーマットです。ASCIIは読み取り値当たりのバイト数が最大なので、読み取り速度が重要でない場合に使用してください。

注記

ASCIIフォーマットを使用する場合は、行シーケンスの終わりのキャリッジ・リターン、改行(**cr**、**lf**)にさらに2バイト必要です。**cr, lf**は、ASCIIフォーマットにだけ使用され、通常は、ASCIIフォーマットの各読み取り値出力の後に続きます。ただし、**ASCII**出力フォーマットを使用していて、RMEMコマンドを使って複数の読み取り値が読み取り値メモリからリコールされた場合には、マルチメータは読み取り値と読み取り値の間にカンマを挿入します(カンマ = 1バイト)。この場合、**cr, lf**は1回だけ発生し、リコールされているグループの最後の読み取り値の後に挿入されます。読み取り値がバスに直接出力された場合(読み取り値メモリがオフの場合)、読み取り値が暗黙の読み取り法を使ってリコールされた場合、またはその他の出力フォーマットを使ってリコールされた場合には、カンマは使用されません。

- **単精度整数(SINT)または倍精度整数(DINT)** 固定レンジ(オートレンジをオフ)で、最高速度の低分解能測定(3.5桁または4.5桁)を行なう場合は、SINTフォーマットを使用します。(SINTフォーマットは読み取り値当たり2バイトに過ぎないので、この他のいずれのフォーマットよりも、SINTを使った方がより高速に、読み取り値をGPIBを経由して転送することができます)。固定レンジでの高速・高分解能測定(5.5.桁以上)には、DINTフォーマットを使用します。

注記

SINTまたはDINTメモリ/出力フォーマットを使用した場合、マルチメータはスケール・ファクタを読み取り値に適用します。スケール・ファクタは、マルチメータの測定ファンクション、レンジ、A/Dコンバータの設定、オンになっている演算に基づきます。SINTまたはDINTフォーマットは、周波数/周期測定(リアルタイム/後処理演算がオンになっている場合(STATまたはPFAILを除く)、またはオートレンジがオンになっている場合)には使用しないでください。

- **単精度実数(SREAL)または倍精度実数(DREAL)** SINTおよびDINTフォーマットとは違い、SREALまたはDREALフォーマットでの読み取り値の出力は、スケーリングされていないので、どのような測定ファンクション/マルチメータ構成ででも使用できます。(スケール・ファクタがないので、SREALおよびDREALフォーマットは、オートレンジ/演算機能がオンになっている場合に適しています)。DREALフォーマットの場合は、コントローラによる変換が一切不要であるという利点もあります。6.5桁以下の分解能での測定には、SREALフォーマットを使用します。6.5桁より大きな分解能での測定には、DREALフォーマットを使用します。

OFORMATコマンドは、読み取り値の出力フォーマットを指定します(電源投入時のデフォルト・フォーマットはASCIIです)。例えば、倍精度整数フォーマットを選択するには、以下を送信します。

```
OUTPUT 722;"OFORMAT DINT"
```

過負荷表示

本器は、次のように、特定の出力フォーマットの最大の数を出力することによって、過負荷状態を知らせます(現在のレンジを上回る入力を測定できます)。

```
SINTフォーマット: +32767 または -32768 (スケーリングされていない)
DINTフォーマット: +2.147483647E+9 または -2.l47483648E+9 (スケーリングされていない)
ASCII、SREAL、DREAL: +/-l.OE+38
```

出力終端

通常は、ASCIIフォーマットでのGPIBへの読み取り値の出力にはそれぞれ、*cr lf*(キャリッジ・リターン、改行)が続きます。*cr lf*は、ほとんどのコントローラに対して伝送の終わりを指示します。他のフォーマットの読み取り値出力には、行シーケンスの終わりに*cr lf*がありません。いずれの出力フォーマットでも、GPIB EOI(End Or Identify = EOI信号)機能をオンにして、伝送の終わりにマークを付けることができます。詳細については、第6章の「ENDコマンド」を参照してください。

SINTまたはDINT出力フォーマットの使用方法

ISCALE?コマンドは、SINTまたはDINTフォーマットの読み取り値出力のスケール・ファクタ(ASCIIフォーマット)を返します。(コントローラがスケール・ファクタを取り出した後は、出力フォーマットは指定したSINTまたはDINTフォーマットに戻ります)。スケール・ファクタは、マルチメータの構成が済んでから読み取りがトリガされるまでの間、またはすべての読み取りが完了し、コントローラに転送されるまでの間取り出すことができます。(ISCALE? コマンドの実行時に読み取り値が出力バッファにある場合には、読み取り値はスケール・ファクタによって上書きされます)。

SINTの例

次のプログラムは、10個の読み取り値をSINTフォーマットで出力し、スケール・ファクタを取り出し、スケール・ファクタに各読み取り値を掛けます。読み取り値は、TRANSFER文 (このコマンドは、BASlC言語を使用するHewlett-Packard 200/300

コントローラに固有のコマンドです)を使って、コントローラに転送されます。TRANSFER文は、GPIBをダイレクト・メモリ・アクセス(DMA) GPIBインタフェースと併用している場合には特に、GPIBを経由して読み取り値を転送するための最も高速な方法です。測定/転送速度が重要な場合には必ず、TRANSFER文を使用してください。

```
10 OPTION BASE 1                          !配列の番号付けは1から開始
20INTEGER Num_readings                    !変数を宣言
30INTEGER Int_rdgs (1: 10) BUFFER         !整数バッファ配列を作成
40REAL Rdgs(l:l0)                         !実数配列を作成
50Num_readings=10                         !読み取り回数＝10
60ASSIGN @Dvm TO 722                      !マルチメータのアドレスを割り当て
70 ASSIGN Int_rdgs TO BUFFER Int_rdgs(*)  !バッファのI/Oパス名を割り当て
80 OUTPUT @Dvm;"PRESET NORM;OFORMAT SINT;NPLC 0;NRDGS ";Num_readings
85   !TARM AUTO、TRIG SYN、SINT出力フォーマット、最小積分時間
90TRANSFER @Dvm TO @Int_rdgs;WAIT         !SYNイベント、読み取り値を
91!整数配列に転送; コンピュータの整数フォーマットはSINTと同じなので、
95!ここではデータ変換は不要(整数配列は必要)
100OUTPUT @Dvm;"I SCALE?"                 !SINTフォーマットのスケール・ファクタを問合せ
110ENTER @Dvm;S                           !スケール・ファクタを入力
120FOR I=1 TO Num_readings
130Rdgs(I)=Int_rdgs(I)                    !各整数読み取り値を実数フォーマットに変換
135 !(次行で整数のオーバーフローの発生を防ぐために必要)
140R=ABS(Rdgs(I))                         !絶対値を使ってOVLDをチェック
150IF R>=32767 THEN PRINT "OVLD"          !OVLDの場合、過負荷メッセージを出力
160Rdgs(I)=Rdgs(I)*S                      !読み取り値にスケール・ファクタを乗算
170Rdgs(I)=DROUND(Rdgs(I),4)              !4桁に丸める
180NEXT I
190END
```

DINTの例　次のプログラムは、50回読み取りを行ない、DINTフォーマットを使って読み取り値をコンピュータに転送するという点を除いて、前のプログラムと同じです。

```
10OPTION BASE 1                           !配列の番号付けは1から開始
20INTEGER Num_readings,l,J,K              !変数を宣言
30Num_readings= 50                        !読み取り回数＝50
40ALLOCATE REAL Rdgs(l:Num_readings)      !読み取り値用の配列を宣言を作成
50ASSIGN @Dvm TO 722                      !マルチメータのアドレスを割り当て
60ASSIGN @Buffer TO BUFFER[4*Num_readings]!バッファ I/Oパス名を割り当て
70OUTPUT @Dvm;"PRESET NORM;RANGE 10;FORMAT DINT;NRDGS";Num_readings
75TARM AUTO, TRIG SYN,DCV 10V RANGE,DINT OUTPUT FORMAT,NRDGS 50,AUTO
80TRANSFER @Dvm TO @Buffer;WAIT           !SYNイベント、読み取り値を転送
90OUTPUT @Dvm;"1 SCALE?"                  !DINTのスケールを問合せ
100ENTER @Dvm;S                           !スケール・ファクタを入力
110FOR I=1 TO Num_readings
120ENTER @Buffer USING "#,W,W";J,K        !16ビットの2の補数ワードをJとKの
121!各変数に1個ずつ入力(#＝文ターミネータは不要;
125!W＝16ビットの2の補数としてデータを入力)
130Rdgs(I)=(J*65536.+K+65536.*(K<0))      !実数に変換
140R=ABS(Rdgs(I))                         !絶対値を使ってOVLDをチェック
150IF R>2147483647 THEN PRINT "OVLD"      !過負荷が発生した場合は、メッセージを出力
160Rdgs(I)=Rdgs(I)*S                      !スケール・ファクタを適用
170Rdgs(I)=DROUND(Rdgs(I),8)              !変換された読み取り値を丸める
180PRINT Rdgs(I)                          !読み取り値を出力
190NEXT I
200END
```

SREAL出力フォーマットの使用方法

次のプログラムは、SREALフォーマットで出力された10個の読み取り値の変換方法を示しています。

```
10OPTION BASE 1                              !配列の番号付けは1から開始
20INTEGER Num_readings                       !変数を宣言
30Num_readings=10                            !読み取り回数 = 10
40ALLOCATE REAL Rdgs(l:Num_readings)         !読み取り値用の配列を宣言を作成
50ASSIGN @Dvm TO 722                         !マルチメータのアドレスを割り当て
60ASSIGN @Buffer TO BUFFER [4*Num_readings]  !バッファ I/Oパス名を割り当て
70OUTPUT @Dvm;"PRESET NORM;OFORMAT SREAL;NRDGS";Num_readings
75!TRIG SYN、SREAL出力フォーマット、1 PLC、DCV AUTORANGE、10個の読み取り値
80TRANSFER @Dvm TO @Buffer;WAIT              !SYNイベント; 読み取り値を転送
90FOR I=1 TO Num_readings
100ENTER @Buffer USING "#,B";A,B,C,D         !1個の8ビット・バイトを
101!各変数に入力、(# =文ターミネータは不要、B = 1個の8ビット・バイト
105!を入力し、0～255の範囲の整数と解釈)
110S=l!SREALからの読み取り値を変換
120IF A>127 THEN S=-1                        !SREALからの読み取り値を変換
130IF A>127 THEN A=A-128                     !SREALからの読み取り値を変換
140A=A*2- 127                                !SREALからの読み取り値を変換
150IF B>127 THEN A=A+l                       !SREALからの読み取り値を変換
160IF B<=127 THEN B=B+128                    !SREALからの読み取り値を変換
170Rdgs(I)=S*(B*65536.+C*256.+D)*2^(A-23)    !SREALからの読み取り値を変換
180Rdgs(I)=DROUND(Rdgs(I),7)                 !読み取り値を7桁に丸める;
181!OVLD値が必ずl.E+38に丸められるように、SREALに関しては必ずこれを行なうこと
185!(丸めないと、値が多少小さくなる可能性がある)
190IF ABS(Rdgs(I))=l.E+38 THEN               !過負荷が発生した場合:
200PRINT "Overload Occurred"                 !過負荷メッセージを出力
210ELSE                                      !過負荷が発生しなかった場合:
220PRINT Rdgs(I)                             !読み取り値を出力
230END IF
240NEXT I
250END
```

DREAL出力フォーマットの使用方法

次のプログラムでは、DREAL出力フォーマットが用いられています。このフォーマットを使用する場合は、DREALは、コントローラが内部データ・フォーマットとして使用するものと同じフォーマット(8バイト/語)なので、変換は一切不要です。

```
10OPTION BASE 1                              !配列の番号付けは1から開始
20REAL Rdgs(l:l0) BUFFER                     !バッファ配列を作成
30ASSIGN @Dvm TO 722                         !マルチメータのアドレスを割り当て
40ASSIGN @Rdgs TO BUFFER Rdgs(*)             !バッファ I/Oパス名を割り当て
50OUTPUT @Dvm;l'PRESET NORM;NPLC 10;OFORMAT DREAL;NRDGS 10"
55!TRIG SYN、10 PLC、DCV AUTORANGE、DREAL出力フォーマット、10回の読み取り/トリガ
60TRANSFER @Dvm TO @Rdgs;WAIT                !SYNイベント、読み取り値を転送
70FOR I=1 TO 10
80IF ABS(Rdgs(I))=l.E+38 THEN                !過負荷メッセージが発生した場合:
90PRINT "OVERLOAD OCCURRED"                  !過負荷メッセージを出力
100ElSE                                      !過負荷が発生しなかった場合:
110Rdgs(I)=DROUND(Rdgs(I),8)                 !読み取り値を丸める
120PRINT Rdgs(I)                             !読み取り値を出力
130END IF
140NEXT I
150END
```

前のプログラムでは、TRANSFER文を使用して、マルチメータから読み取り値を取

り込みました。次のプログラムでは、ENTER文を用いて、読み取り値をDREALフォーマットでコンピュータに転送します。ENTER文は、I/Oパスが不要なので、TRANSFER文よりも使いやすいのですが、速度は遅くなります。また、ENTERを使用する場合には、FORMAT OFFコマンドを使って、ASCIIの代わりに内部データ構造を使用するようにコントローラに指示する必要があります。

```
10OPTION BASE 1                        !配列の番号付けは1から開始
20Num_readings=20                      !読み取り回数 = 20
30ALLOCATE REAL Rdgs(l:Num_readings)   !読み取り値用の配列を作成
40ASSIGN @Dvm TO 722                   !マルチメータのアドレスを割り当て
50OUTPUT @Dvm;"PRESET NORM;OFORMAT DREAL;NPLC 10;NRDGS";Num_readings
55!TRIG SYN、DCV AUTORANGE、DREAL出力フォーマット、10 PLC、20回の読み取り/トリガ
60ASSIGN @Dvm;FORMAT OFF               !8バイト/ワード・データ構造を使用
70FOR I=1 TO Num_readings
80ENTER @Dvm;Rdgs(I)                   !各読み取り値を入力
90IF ABS(Rdgs(I))=l.E+38 THEN          !過負荷が発生した場合:
100PRINT "OVERLOAD OCCURRED"           !過負荷メッセージを出力
110ELSE                                !過負荷が発生しなかった場合
120Rdgs(I)=DROUND(Rdgs(1),8)           !読み取り値を8桁に丸める
130PRINT Rdgs(I)                       !読み取り値を出力
140END IF
150NEXT I
160END
```

読み取り速度を上げる

このセクションでは、マルチメータの高速モードと、読み取り速度に影響を及ぼす要因について説明します。また、サンプル・プログラムを使って、読み取り速度を上げる方法、高速で読み取り値をコントローラに直接転送する方法、読み取り値メモリからコントローラへの高速転送を実行する方法、読み取り速度を決定する方法を説明します。

高速モード

DC電圧測定、DC電流測定、2端子/4端子抵抗測定、直接/サブ・サンプリング測定[1]については、読み取りが開始され、積分時間が10 PLCより短く、さらに以下の各コマンドが実行された場合に、マルチメータは高速モードになります。

```
ARANGE OFF
DISP OFF
MATH OFF
MFORMAT SINTまたはDINT (読み取り値メモリがオンの場合にのみ必要)
OFORMAT SINTまたはDINT (読み取り値メモリがオフの場合にのみ必要)
```

読み取りが高速モードで行なわれている間は、マルチメータは完全に測定プロセス専用になります。つまり、指定した読み取りが完了するまで、コマンドの処理は一切行なわれません。読み取り値が高速モードで出力バッファに直接送られている場合は、マルチメータは、各読み取り値が出力バッファから削除されるまで待ってから、次の読み取り値を出力バッファに入れます。したがってバス/コントローラの速度の制限により読み取り値が失われることはありません。(高速モードでない場合には、マルチメータは、新しい読み取り値が入手可能になると、出力バッファ内の

1. 直接およびサブサンプリング測定についての詳細は、第5章を参照してください。

読み取り値を上書きします)。

FIFOモードで読み取り値メモリをオンにすると、高速モードで一杯になった場合には、トリガ・アーム・イベントがHOLDになり、読み取りは停止し、マルチメータは高速モードではなくなります。読み取り値の一部または全部がメモリから削除された後に、トリガ・アーム・イベントを変更することによって(TARMコマンド)、測定を再開することができます。LIFOモードでは、読み取り値メモリが一杯になると、高速モードにあるか否かに関わらず、最も古い読み取り値が最新の読み取り値に置き換えられます。

注記

高速モードでは、読み取りが実行されている間は、入力バッファが一時的にオフになります。また、END ALWAYSを指定した場合(GPIB EOIモードを指定した場合)は、読み取りが実行されている間は、EOIモードがEND ONに変わります。読み取りが完了すると、入力バッファ・モードとEOIモードは、前に指定したモードに戻ります。

高速モードでは、マルチメータはGPIB CLEARコマンド(デバイス・クリア)にだけ応答します。何らかの理由でマルチメータを高速モードから解放する必要がある場合には、以下を送信します。

```
CLEAR 722
```

CLEARコマンドは、測定を中断し、マルチメータを高速モードから解放します。GPIB CLEARコマンドの詳細については、付録Bを参照してください。

高速読み取り用の構成

PRESET FASTコマンドは、高速読み取り用に構成するための一連のコマンドを実行します。さらに、読み取り速度は、積分時間や分解能、トリガ設定、遅延時間、AC帯域幅(AC測定の場合のみ)に左右されます。さらに、抵抗測定についてだけは、オフセット補正モードの影響も受けます。

注記

このセクションで説明するコマンド以外にも、DEFEATコマンドを使って、マルチメータの入力保護アルゴリズムやいくつかのシンタックス/エラー・チェック・アルゴリズムをオフにすることによって、処理能力のスピードアップを図ることができます。これらのアルゴリズムをオフにした場合には、オフにしなかった場合よりも迅速に、マルチメータは新しい測定構成に変化します。詳細および使用法に関する注意については、第6章の「DEFEATコマンド」を参照してください。

PRESET FASTコマンド

PRESET FASTコマンドは、読み取り速度を低下させる多くの機能をオフにして、メモリおよびGPIBへの高速読み取り値転送用にマルチメータを構成します。表22には、PRESET FASTによって実行される速度関連のコマンドと、各コマンドの実行理由が示されています。

表22. PRESET FASTによって実行されるコマンド

コマンド	理由
DCV 10	10VレンジのDC電圧測定を選択します。これによって、オートレンジがオフになります。オートレンジ機能は各読み取りの前に入力をサンプリングするため、固定レンジで行なわれる読み取りよりも読み取り当たりの時間が長くなります。固定レンジの欠点は、フルスケールの10%未満の信号の分解能が低いことと、フルスケールより大きい読み取り値の場合に過負荷条件が発生する可能性があることです。
AZERO OFF	オートゼロがオンの場合、各読み取りに続いてゼロ測定が実行されるため(DC測定の場合のみ)、読み取り当たりの時間が増加します。
DISP OFF	マルチメータがディスプレイを更新するのに要する時間により、読み取り速度が低下します。
MATH OFF	オンになっているリアルタイム演算により、読み取り速度が低下します。読み取り値に関する演算を実行する必要がある場合には、後処理演算(MMATH コマンド)を使用します。詳細については、本章で後述の「演算」を参照してください。
MFORMAT DINT	読み取り値は、A/DコンバータからSINTまたはDINTフォーマットで送られます(使用されるフォーマットは、指定した測定分解能によって決まります。* PRESET FASTによって選択された構成では、A/DコンバータはDINTを使用します)。読み取り値をメモリに転送するための最も高速の方法は、メモリ・フォーマット(MFORMAT)とA/Dコンバータのフォーマットを一致させて、変換しなくて済むようにすることです。(SINTまたは DINTを使用すべき場合については、本章で前述の「読み取り値のフォーマット」を参照してください)。
OFORMAT DINT	読み取り値は、A/DコンバータからSINTまたはDINTフォーマットで送られます(使用されるフォーマットは、指定した測定分解能によって決まります。* PRESET FASTによって選択された構成では、A/DコンバータはDINTを使用します)。読み取り値を出力バッファに転送するための最も高速の方法は、出力フォーマット(MFORMAT)とA/Dコンバータのフォーマットを一致させて、変換しなくて済むようにすることです。さらに、出力フォーマットと読み取り値メモリ・フォーマットが一致している場合には、メモリから読み取り値をリコールするのに変換は不要です。SINTまたはDINTフォーマットを使用している場合には、ISCALE? コマンドを使ってスケール・ファクタを取り出すことを忘れないでください。(SINTまたは DINTを使用すべき場合については、本章で前述の「読み取り値のフォーマット」を参照してください)。

* 直接サンプリングしたディジタイズの場合、使用されるフォーマットは、入力信号の振幅によって決まります。詳細については、第5章を参照してください。

積分時間と分解能

DC、抵抗およびアナログAC測定: 指定した積分時間または分解能は、DC電圧、DC電流、2端子/4端子抵抗、AC/AC+DC電流、ACまたはAC+DC電圧(SETACV ANA法だけを使用)の読み取り速度に重大な影響を及ぼします。積分時間が長いほど(すなわち、分解能が大きいほど)、読み取り速度は遅くなります。付録Aの仕様には、積分時間に基づいた各測定の読み取り速度が示されています。

サンプリングAC電圧測定: SETACV SYNCまたはSETACV RNDMを使ったAC/AC+DC測定の場合、積分時間は一定で、変更できません。これらの測定については、指定した分解能が読み取り速度に重大な影響を及ぼします。付録Aの仕様には、指定した分解能に基づいたサンプリングAC測定の読み取り速度が示されています。

周波数/周期測定: 積分時間が周波数/周期測定に影響を及ぼすことはありません。これらの測定については、指定した分解能(これはゲート時間も選択)が読み取り速度に重大な影響を及ぼします。付録Aの仕様には、指定した分解能に基づいた周波数測定と周期測定の読み取り速度が示されています。

トリガ設定

最高速のトリガ構成を実現するには、トリガ・アーム・イベント、トリガ・イベント、サンプリング・イベントをAUTOに設定します。TIMERサンプリング・イベント(またはSWEEPコマンド)を使用することも可能です。TRIGGER TOO FASTエラーが生じていないと仮定した場合、読み取り速度はTIMERまたはSWEEP間隔の逆数になります。

遅延時間

通常操作では、マルチメータは、現在の測定ファンクション、レンジ、分解能、およびAC帯域幅設定(AC測定の場合)に基づいて、遅延時間(デフォルト遅延)を自動的に決定します。この遅延時間は、実際には、最初の読み取りの前に挿入されるセトリング時間で、これにより確度の高い測定を実現します。デフォルト遅延は、AC測定の読み取り速度に大きな影響を及ぼしますが、サンプリングAC電圧またはDC測定の読み取り速度に及ぼす影響はごくわずかです。アナログAC測定については、デフォルト値より短い遅延を指定することによって、より高速の読み取り速度を実現できます。ただし、その結果として生じるセトリング時間は、確度の高い測定の実現を不可能にする可能性があります。

AC帯域幅

最高速のAC測定を実現するためには、入力信号の周波数成分と一致するAC帯域幅を指定します(ACBANDコマンド)。付録Aの仕様には、入力信号の周波数成分に基づいたAC測定の読み取り速度が示されています。

オフセット補正

オフセット補正をオンにした2端子/4端子抵抗測定の場合、オフセット電圧の測定は、各抵抗測定の前に行なわれます。このため、オフセット補正をオフにした場合(OCOMP OFF)よりも時間が掛かります。

高速DCVの例

次のプログラムは、最高速度(1秒当たり100000回以上の読み取り)でDC電圧を測定します。読み取り値は読み取り値メモリに保存されます。

```
10 OUTPUT 722;"PRESET FAST"          !DCV、10Vレンジ、TARM SYN、TRIG AUTO
20 OUTPUT 722;"APER 1.4E-6"          !最大積分時間
25                                   !100000回/秒以上の読み取り
30 OUTPUT 722;"MFORMAT SINT"         !SINTメモリ・フォーマット
40 OUTPUT 722;"MEM FIFO"             !読み取り値メモリをオン
50 OUTPUT 722;"NRDGS 10000,AUTO      !10000回の読み取り/トリガ、自動サンプリング・イベント
60 OUTPUT 722;"TARM SGL"             !読み取りをトリガ
70 END
```

高速OHM(またはOHMF)の例

次のプログラムは、最高速度(1秒当たり100000回以上の読み取り)で2端子抵抗を測定します。このプログラムは、ライン50のOHMコマンドの代わりにOHMFコマンドを用いることによって、4端子抵抗にも適用できます。

```
10 OUTPUT 722;"PRESET FAST"          !DCV 10Vレンジ、TARM SYN、TRIG AUTO
20 OUTPUT 722;"APER 1.4E-6"          !>1OO000回の読み取り/秒が可能な
25                                   !最大積分時間
30 OUTPUT 722;"MFORMAT SINT"         !SINTメモリ・フォーマット
40 OUTPUT 722;"MEM FIFO"             !読み取り値メモリをオン
50 OUTPUT 722;"OHM 100E3"            !2端子抵抗、100K(Ω)レンジ
60 OUTPUT 722;"NRDGS 10000,AUTO"     !10000回の読み取り/トリガ、自動サンプリング・イベント
```

```
70 OUTPUT 722;"TARM SGL"            !読み取りをトリガ
80 END
```

高速DCIの例　次のプログラムは、DC電流を高速測定します。

```
10 OUTPUT 722;"PRESET FAST"         !DCV、10Vレンジ、TARM SYN、TRIG AUTO
20 OUTPUT 722;"APER 1.4E-6"I        !最高読み取り速度が可能な
25                                  !最大積分時間
30 OUTPUT 722;"MFORMAT SINT"        !SINTメモリ・フォーマット
40 OUTPUT 722;"MEM FIFO"            !読み取り値メモリをオン
50 OUTPUT 722;"DCI 100E-3"          !DC電流、100mAレンジ
60 OUTPUT 722;"NRDGS 5000 AUTO"     !5000回の読み取り/トリガ、自動サンプリング・イベント
70 OUTPUT 722;"TARM SGL"            !読み取りをトリガ
80 END
```

高速同期ACV/ACDCVの例　次のプログラムは、最高速度(1秒当たり約10回の読み取り)で、同期法を使ってAC電圧を測定します。このプログラムは、ライン50のACVコマンドの代わりにACDCVコマンドを用いることによって、AC+DC電圧にも適用できます。

```
10 OUTPUT 722;"PRESET FAST"         !TARM SYN、TRIG AUTO
20 OUTPUT 722;"MFORMAT SINT"        !SINTメモリをフォーマット
30 OUTPUT 722;"MEM FIFO"            !読み取り値メモリをオン
40 OUTPUT 722;"SETACV SYNC"         !同期AC測定法
50 OUTPUT 722;"ACV 10,2"            !AC電圧、10Vレンジ、2%分解能
60 OUTPUT 722;"ACBAND 5E3,8E3"      !5kHz～8kHzの信号
70 OUTPUT 722;"NRDGS 20, AUTO"      !20回の読み取り/トリガ、自動サンプリング・イベント
80 OUTPUT 722;"TARM SGL"            !読み取りをトリガ
90 END
```

高速ランダムACV/ACDCVの例　次のプログラムは、最高速度(1秒当たり約45回の読み取り)で、ランダム法を使ってAC電圧を測定します。このプログラムは、ライン50のACVコマンドの代わりにACDCVコマンドを用いることによって、AC+DC電圧にも適用できます。

```
10 OUTPUT 722;"PRESET FAST""       !TARM SYN、TRIG AUTO
20 OUTPUT 722;"MFORMAT SINT        !SINTメモリ・フォーマット
30 OUTPUT 722;"MEM FIFO"           !読み取り値メモリをオフ
40 OUTPUT 722;"SETACV RNDM"        !ランダムAC測定方法
50 OUTPUT 722;"ACV 10 6"           !AC電圧、10Vレンジ、6%分解能
60 OUTPUT 722;"ACBAND 10E3,20E3″   !10kHz～20kHzの信号
70 OUTPUT 722;"NRDGS 100, AUTO″    !100回の読み取り/トリガ、自動サンプリング・イベント
80 OUTPUT 722;"TARM SGL"           !読み取りをトリガ
90 END
```

高速アナログACV/ACDCVの例　次のプログラムは、アナログ法を使ってAC電圧を高速測定します。このプログラムでは、デフォルトの遅延時間が用いられています。より短い遅延時間を指定することによって、より高速の読み取り速度を実現できます。ただし、その結果として生じるセトリング時間は、確度の高い測定の実現を不可能にする可能性があります。ライン60でより短い積分時間を指定することによって、仕様化されていないより高速の読み取り速度を実現することも可能です。このプログラムは、ライン50のACVコマンドの代わりにACDCVコマンドを用いることによって、AC+DC電圧にも適用できます。

```
10 OUTPUT 722;"PRESET FAST""       !TARM SYN、TRIG AUTO
20 OUTPUT 722;"MFORMAT SINT        !SINTメモリをフォーマット
30 OUTPUT 722;"MEM FIFO"           !読み取り値メモリをオン、FIFOモード
40 OUTPUT 722;"SETACV ANA"         !アナログAC測定法
```

```
50 OUTPUT 722;"ACV 10"              !AC電圧、10Vレンジ
60 OUTPUT 722;"NPLC 0.1""           !0.1 PLCの積分時間
70 OUTPUT 722;"ACBAND 10E3,20E3"    !10kHz～20kHzの信号
80 OUTPUT 722;"NRDGS 100, AUTO"     !100回の読み取り/トリガ、自動サンプリング・イベント
90 OUTPUT 722;"TARM SGL"            !読み取りをトリガ
100 END
```

高速ACI/ACDCIの例

次のプログラムは、AC電流を高速測定します。このプログラムでは、デフォルトの遅延時間が用いられています。より短い遅延時間を指定することによって、より高速の読み取り速度を実現できます。ただし、その結果として生じるセトリング時間は、確度の高い測定の実現を不可能にする可能性があります。ライン50でより短い積分時間を指定することによって、仕様化されていないより高速の読み取り速度を実現することも可能です。このプログラムは、ライン40のACIコマンドの代わりにACDCIコマンドを用いることによって、AC+DC電流にも適用できます。

```
10 OUTPUT 722;"PRESET FAST"         !TARM SYN、TRIG AUTO
20 OUTPUT 722;"MFORMAT SINT"        !SINTメモリ・フォーマット
30 OUTPUT 722;"MEM FIFO"            !読み取り値メモリをオン、FIFOモード
40 OUTPUT 722;"ACI 100E-3"          !AC電流、100mVレンジ
50 OUTPUT 722;"NPLC 0.1"            !0.1 PLCの積分時間
60 OUTPUT 722;"ACBAND 10E3,20E3"    !10kHz～20kHzの信号
70 OUTPUT 722;"NRDGS 100,AUTO"      !100回の読み取り/トリガ、自動サンプリング・イベント
80 OUTPUT 722;"TARM SGL"            !読み取りをトリガ
90 END
```

高速FREQ(またはPER)の例

次のプログラムは、周波数を高速測定します。このプログラムは、ライン40のFREQコマンドの代わりにPERコマンドを用いることによって、周期測定にも適用できます。

```
10 OUTPUT 722;"PRESET FAST"         !TARM SYN、TRIG AUTO
20 OUTPUT 722;"MFORMAT SREAL"       !単精度実数メモリ・フォーマット
30 OUTPUT 722;"MEM FIFO"            !読み取り値メモリをオン、FOFOモード
40 OUTPUT 722;"FREQ 10, .1"         !周波数、10Vレンジ、100μsのゲート時間
50 OUTPUT 722;"ACBAND 10E3,20E3"    !10kHz～20kHzの信号
60 OUTPUT 722;"NRDGS 100, AUTO"     !100回の読み取り/トリガ、自動サンプリング・イベント
70 OUTPUT 722;"TARM SGL"            !読み取りをトリガ
80 END
```

GPIBによる高速転送

出力フォーマットをA/Dコンバータが使用するフォーマットと一致するようにようように設定することによって(SINTまたはDINT)、読み取り値のコントローラへの高速転送が実現します。これは、マルチメータでフォーマット変換が不要であるためです。固定レンジでの高速、低分解能測定（3.5桁または4.5桁)には、SINT出力フォーマットを使用します。(SINTフォーマットは読み取り値当たり2バイトを占有するだけなので、この他のいずれのフォーマットよりも、SINT出力フォーマットを使った方がより高速に、複数の読み取り値をバスを経由して転送できます)。固定レンジでの高分解能読み取り値(5.5.桁以上)の高速転送には、DINT出力フォーマットを使用します。

本器は、読み取りを行なって、1秒当たり100000回以上の読み取り速度で、読み取り値をコントローラに出力することができます。この読み取り速度でSINT出力フォーマットを使用した場合、GPIBとコントローラは1秒当たり200000バイト以上の読み取り速度でデータを転送できなければなりません。Hewlett-Packard シリーズ200/300コンピュータの場合、これにはダイレクト・メモリ・アクセス(DMA)カードが必要です。さらに、最高の転送速度を実現するためには、GPIBバスの動作を遅くさせるデバイスや不必要な長さのGPIBケーブルも取り除く必要があります。

次のプログラムは、最高速度で、読み取り値を直接コントローラに転送します。このプログラムは、最高1秒当たり>100000回の読み取り速度で測定を行ないます。読み取り値は、SINTフォーマットで出力されます。バス/コントローラが>200000バイト/秒の速度で読み取り値を転送できない場合には、読み取り速度は低下します。これは、高速モードでは、マルチメータは、各読み取り値が出力バッファから削除されるのを待ってから、次の読み取り値を出力バッファに入れるためです。次のプログラムでは、SYNトリガ・アーム・イベントを使って、読み取りがトリガされています(TRIG SYNも使用できます)。SYNイベントは、コントローラがマルチメータによる最初の読み取り値の出力を受け入れる準備が整ったことを保証するので、高速処理には非常に重要です。TRANSFER文(ライン120)は、SYNイベントを指定するもので、GPIBをダイレクト・メモリ・アクセス(DMA) GPIBインタフェースと併用している場合には特に､GPIBを経由して読み取り値を転送するための最も高速な方法です。

```
10 OPTION BASE 1                          !配列の番号付けは1から開始
20 INTEGER Num_readings                   !変数を宣言
30 INTEGER Int_rdgs(l:30000) BUFFER       !バッファ用の整数配列を作成
40 REAL Rdgs(l:30000)                     !実数配列を作成
50 Num_readings=30000                     !読み取り回数 = 30000
60 ASSIGN @Dvm TO 722                     !マルチメータのアドレスを割り当て
70 ASSIGN Int_rdgs TO BUFFER Int_rdgs(*)  !バッファ I/Oパス名を割り当て
80 OUTPUT @Dvm; "PRESET FAST"             !TARM SYN、TRIG AUTO、DCV 10V
90 OUTPUT @Dvm;"APER 1.4E-6"              !1.4µsの積分時間
100 OUTPUT @Dvm; "OFORMAT SINT"           !SINT出力フォーマット
110 OUTPUT @Dvm; "NRDGS"; Num_readings    !30000回の読み取り/トリガ、自動
115                                       !サンプリング・イベント(デフォルト値)
120 TRANSFER @Dvm TO @Int rdgs;WAIT       !SYNイベント、読み取り値を
121 !整数配列に転送; コンピュータの整数フォーマットはSINTと同じなので、
125 !ここではデータ変換は不要(整数配列は必要)
130 OUTPUT @Dvm; "ISCALE?"                !SINTフォーマットのスケール・ファクタを問合せ
140 ENTER @Dvm;S                          !スケール・ファクタを入力
150 FOR I=1 TO Num_readings
160 Rdgs(I)=Int_rdgs(I)                   !各整数読み取り値を実数フォーマットに変換
165!(次行で整数のオーバーフローの発生を防ぐために必要)
170 R=ABS(Rdgs(I))                        !絶対値を使ってOVLDをチェック
180 IF R>=32767 THEN PRINT "OVLD"         !OVLDの場合、過負荷メッセージを出力
190 Rdgs(I)=Rdgs(I)*S                     !読み取り値にスケール・ファクタを乗算
200 Rdgs(I)=OROUND(Rdgs(I),4)             !4桁に丸める
210 NEXT I
220 END
```

メモリからの高速転送

読み取り値メモリ・フォーマット(MFORMATコマンド)を出力フォーマット(OFORMATコマンド)と一致するように設定することによって、読み取り値メモリからコントローラへの読み取り値の最高速転送に確実に対応するようにできます。これは、メモリから読み取り値がリコールされる際の変換が不要であるためです。固定レンジでの高速・低分解能測定(3.5桁または4.5桁)には、SINTフォーマットを使用します。(SINTフォーマット読み取り値当たり2バイトを占有するだけなので、この他のいずれのフォーマットよりも､SINT出力フォーマットを使った方がより高速に､複数の読み取り値をメモリに保存し、バスを経由して転送することができます)。固定レンジでの高分解能読み取り値(5.5.桁以上)の高速転送には、DINTフォーマットを使用します。オートレンジがオンで、転送速度が重要な場合には必ず、SREALフォーマット(6.5桁未満の読み取り値の場合)かDREALフォーマット(7.5桁か8.5桁の読み取り値の場合)を使用します。ディスプレイと演算をオフにすることによっても、読み取り値メモリからコントローラへの高速転送を実現します。

次のプログラムは、最高速度でのメモリからコントローラへの読み取り値の転送の一例を示したものです。このプログラムは、SINTフォーマットを使って、読み取り値メモリに5000個の読み取り値を保存します。読み取り値は、「暗黙の読み取り」法を使ってメモリから削除され、TRANSFER文(ライン130)を使ってコントローラに転送されます(SINTフォーマット)。コントローラは次に、スケール・ファクタを取り出し、スケール・ファクタに各読み取り値を掛け、補正した読み取り値を*Rdgs*配列に保存します。

```
10 OPTION BASE 1                                    !配列の番号付けは1から開始
20 INTEGER Num_readings                             !変数を宣言
30 INTEGER Int_rdgs(l:30000) BUFFER                 !バッファ用の整数配列を作成
40 REAL Rdgs(l:30000)                   !実数配列を作成
50 Num_readings=30000                   !読み取り回数 = 30000
60 ASSIGN @Dvm TO 722                   !マルチメタのアドレスを割り当て
70 ASSIGN Int_rdgs TO BUFFER Int_rdgs(*)  !バッファ I/Oパス名を割り当て
80 OUTPUT @Dvm; "PRESET FAST"           !TARM SYN、TRIG AUTO、DCV 10V
90 OUTPUT @Dvm;"APER 1.4E-6"            !1.4µsの積分時間
100 OUTPUT @Dvm; "OFORMAT SINT"         !SINT出力フォーマット
110 OUTPUT @Dvm; "NRDGS"; Num_readings  !30000回の読み取り/トリガ、自動
115                                     !サンプリング・イベント(デフォルト値)
120 TRANSFER @Dvm TO @Int rdgs;WAIT     !SYNイベント、読み取り値を
121 !整数配列に転送; コンピュータの整数フォーマットはSINTと同じなので、
125 !ここではデータ変換は不要(整数配列は必要)
130 OUTPUT @Dvm; "ISCALE?"              !SINTフォーマットのスケール・ファクタを問合せ
140 ENTER @Dvm;S                        !スケール・ファクタを入力
150 FOR I=1 TO Num_readings
160 Rdgs(I)=Int_rdgs(I)                 !各整数読み取り値を実数フォーマットに変換
165!(次行で整数のオーバーフローの発生を防ぐために必要)
170 R=ABS(Rdgs(I))                      !絶対値を使ってOVLDをチェック
180 IF R>=32767 THEN PRINT "OVLD"       !OVLDの場合、過負荷メッセージを出力
190 Rdgs(I)=Rdgs(I)*S                   !読み取り値にスケール・ファクタを乗算
200 Rdgs(I)=OROUND(Rdgs(I),4)           !4桁に丸める
210 NEXT I
220 END
```

読み取り速度の決定

TIMERサンプリング・イベントまたはSWEEPコマンドを使用する場合は、読み取り速度は指定した読み取り値間の間隔の逆数に過ぎません(TRIGGER TOO FASTエラーが生じていないと仮定した場合)。例えば、TIMER間隔をlE-4と指定した場合、読み取り速度は1/l E-4 = 10,000回/秒になります。別のサンプリング・イベントを使用する場合は、トリガ当たりの読み取り回数を多く設定して、各読み取りの後の出力パルスを指定し(EXTOUT RCOMPコマンド)、電子式周波数カウンタをマルチメータの**Ext Out**コネクタに接続することによって、読み取り速度を決定することが可能です。カウンタに表示される周波数は、1秒当たりの読み取り回数で表された読み取り速度です。

コントローラを使って、TARM SGLまたはTRIG SGLコマンドによって開始された多数の読み取りの時間の長さを決定する方法もあります。入力バッファがオフになっていると(INBUF OFF)、SGLイベントは、読み取りがすべて完了するまで、GPIBバスをホールド状態にします。つまり、TARM SGLまたはTRIG SGLコマンドの実行に必要な時間が測定の合計時間になります。例えば、次のプログラムは、読み取り値を読み取り値メモリに保存し、10000回の読み取りのTARM SGLの時間の長さを決定し、10000で合計時間を割って、1秒当たりの読み取り回数を表示します。

TIMEDATEコマンド(ライン90および110)は、BASIC言語を使用するHewlett-Packardシリーズ200/300コンピュータに適用されます。コンピュータのタイマの詳細な使用方法については、コンピュータの操作マニュアルを参照してください。

```
10 REAL Num_readings                            !配列を作成
20 Num_readings=10000                           !読み取り回数 = 10000
30 ASSIGN @Dvm to 722                           !マルチメータのアドレスを割り当て
40 OUTPUT @Dvm;"PRESET FAST"                    !DCV 10Vレンジ、DINTメモリ・フォーマット、高速
45                                              !読み取り、TARM SYN、TRIG AUTO
50 OUTPUT @Dvm;"NPLC 0"                         !最小積分時間(500ns)
60 OUTPUT @Dvm;"MEM FIFO"                       !読み取り値メモリをオン、FIFOモード
70 OUTPUT @Dvm;"MFORMAT SINT"                   !SINTメモリ・フォーマット
80 OUTPUT @Dvm;"NRDGS"; Num_readings ,"AUTO"    ! 10000回の読み取り/トリガ、自動
85                                              !サンプリング・イベント
90 TO=TIMEDATE                                  !タイマを開始
100 OUTPUT @Dvm;"TARM SGL"                      !読み取りをトリガ
110 T1=TIMEDATE                                 !タイマを停止
120 PRINT "Readings per second = ";Num_readings/(T1-T0)
125                                             !読み取り値/秒を出力
130 END
```

読み取り値メモリを使う代わりに、バスを経由して複数の読み取り値を転送する場合には、SYN(同期)トリガ・アームまたはトリガ・イベント(読み取りがすべて完了し、転送されるまで、バスをホールド状態にします)を使用して、コントローラのENTERまたはTRANSFER文の時間の長さを決定することができます。これについては、次のプログラムを参照してください(同期トリガ・アーム・イベントは、ライン50のPRESET FASTコマンドによって選択されます)

```
10 REAL Num_readings                            !配列を作成
20 Num_readings=300000                          !読み取り回数 = 300000
30 ASSIGN @Dvm TO 722                           !マルチメータのアドレスを割り当て
40 ASSIGN @Buffer TO BUFFER [2*Num_readings]    !バッファ I/Oパス名を割り当て
50 OUTPUT @Dvm; "PRESET FAST"                   !DCV 10Vレンジ、DINT出力フォーマット、
55                                              !TARM SYN、TRIG AUTO
60 OUTPUT @Dvm;"NPLC 0"                         !最小積分時間
70 OUTPUT @Dvm;"OFORMAT SINT"                   !SINT出力フォーマット
80 OUTPUT @Dvm; "NRDGS "; Num_readings, "AUTO"
85 !300000回の読み取り/トリガ、自動サンプリング・イベント
90 TO=TIMEDATE                                  !タイミング読み取りを開始
100 TRANSFER @Dvm TO @Buffer;WAIT               !SYNイベント、読み取り値を転送
110 T1=TIMEDATE                                 !タイミング読み取りを停止
120 PRINT "READINGS PER SECOND = 11;Num_readings/(T1/T0)
125                                             !読み取り値/秒を出力
130 END
```

注記 スケール・ファクタ(SINTフォーマットで出力された読み取り値の変換に必要)の取り出しに必要な時間は、上のプログラムには含まれていません。

EXTOUT信号

指定のA/Dコンバータ・イベントが発生した場合、マルチメータがGPIBサービス・リクエストを生成した場合、またはEXTOUT ONCEコマンドが実行された場合に、

Ext OutコネクタにTTL互換信号を出力するように､マルチメータをプログラムすることができます。この信号を用いることによって、外部機器をマルチメータと同期させることができます。EXTOUTコマンドの最初のパラメータは、この信号を生成するイベントを指定し、2番目のパラメータは、信号の極性(NEG = 立下がり、POS = 立上がり)を指定します。**Ext Out**コネクタ上に信号を生成することのできるイベントを以下に示します。

- 読み取り完了
- バースト読み取り完了
- 入力完了
- アパーチャ波形
- サービス・リクエスト
- EXTOUT ONCEコマンド実行

上に示されているイベントのほとんどは、マルチメータのA/Dコンバータに適用できます。図20は、これらのイベントとA/Dコンバータの動作との関係を示したものです。

注記　図20に示されている外見上の時間間隔は、図示することだけを目的としており、マルチメータによって生成される実際の間隔を示すためのものではありません。

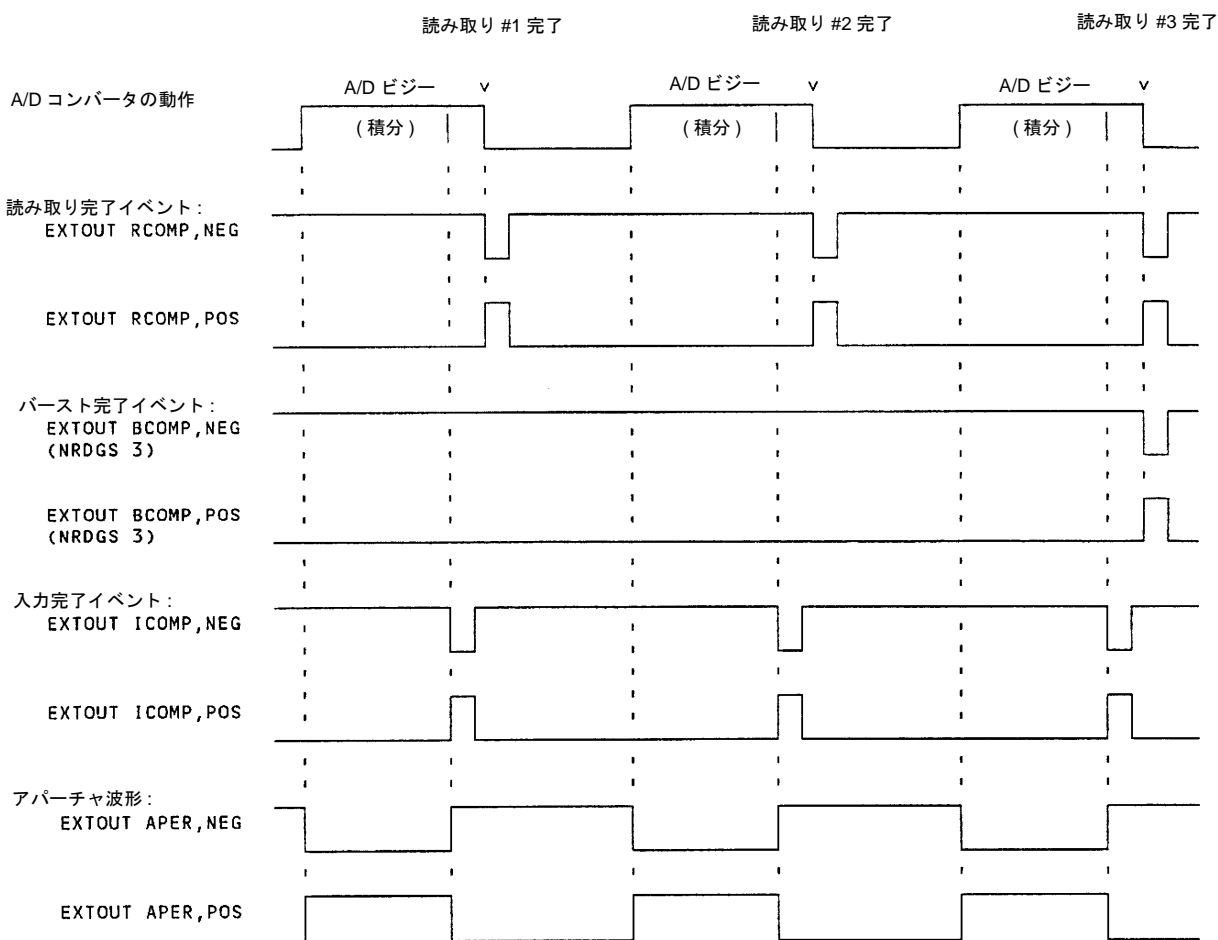



図20. A/Dコンバータとイベントの関係

読み取り完了

指定した場合には、読み取り完了イベント(RCOMPイベント)は、いずれの測定ファンクションの場合にも、各読み取りに続いて1 μsのパルスを生成します。サンプリングAC電圧測定(SETACV SYNCまたはRNDM)の場合、測定プロセスの各サンプルの後ではなく、計算された各読み取り値の後に、パルスが出力されます。このイベントを用いることによって、スキャナ・チャネル当たり1回の読み取りを実行する場合に、外部スキャナをマルチメータと同期させることができます。

次のプログラムでは、RCOMPイベントを使って、マルチメータとスキャナの同期が図られています(例では、スロット200にスキャン・モジュールが搭載された3235スイッチ/テスト・ユニットが用いられています)。測定用の接続については、図21を参照してください。スキャナは、各チャネル閉鎖後に立下がりパルスを出力するようにプログラムされています(ライン60)。このパルスは、マルチメータの**Ext Trig**コネクタに接続され、各読み取りをトリガします。各読み取りの後に、マルチメータのEXTOUT信号によって、スキャナは次のチャネルに進みます。チャネル閉鎖によって、信号が生成され、それによって次の読み取りがトリガされます。このシーケンスは、6つのチャネルすべてがスキャンされるまで繰り返されます。読み取り値は、マルチメータの読み取り値メモリの保存されます。

```
10 OUTPUT 722;"PRESET NORM"          !DCV、NRDGS、l、AUTO、TARM AUTO、TRIG SYN
20 OUTPUT 722;"MEM FIFO"             !読み取り値メモリをオン、FIFOモード
30 OUTPUT 722;"TRIG EXT"             !トリガ・イベント = 外部
40 OUTPUT 722;"EXTOUT RCOMP,NEG"     !読み取り完了EXTOUT、立下がりTTL
45                                   !外部スキャナを構成
50 OUTPUT 709;"SADV EXTIN"           !マルチメータのEXTOUT信号でスキャナを次へ進める
60 OUTPUT 709;"CHCLOSED EXT"         !各チャネル閉鎖後に立下がりパルスを出力
70 OUTPUT 709;"SCAN 201- 206"        !スロット200のスキャナでチャネル01～06をスキャン
75                                   !さらに、チャネル01に進み、スキャンを開始
80 END
```

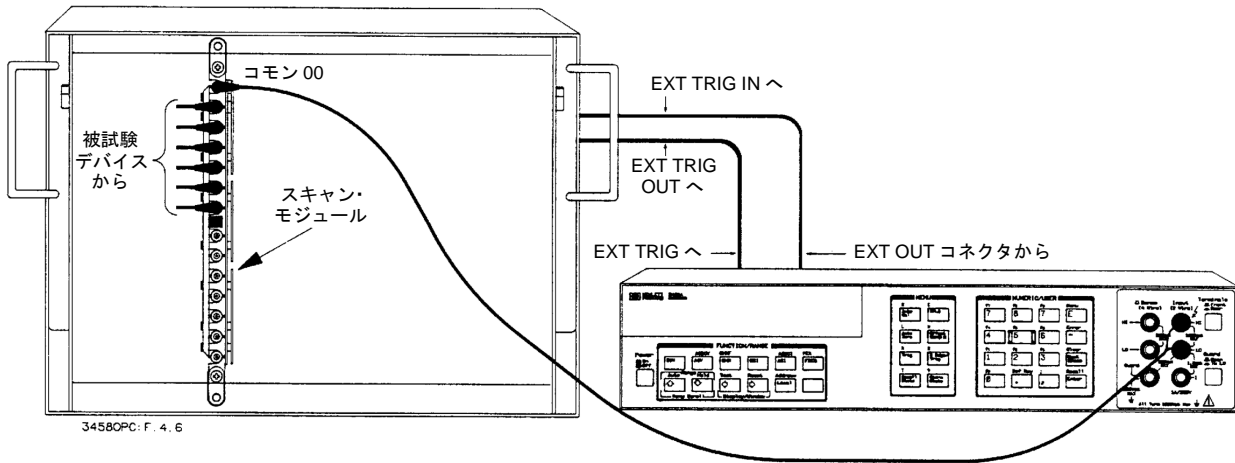



図21. 外部スキャナの使用方法

バースト完了

指定した場合には、バースト完了イベント(BCOMPイベント)は、1つのグループの読み取りの完了に続いて、1 μsのパルスを生成します。1グループ当たりの読み取り回数は、NRDGSまたはSWEEPコマンドによって指定します。BCOMPイベントを用いることによって、スキャナ・チャネル当たり複数の読み取りを実行する場合に、外部スキャナをマルチメータと同期させることができます。次のプログラムは、BCOMPイベントを使用して、各スキャナ・チャネルで15回の読み取りを実行するということ以外は、前のプログラムと似ています。本例の接続図については、図21を参照してください。

```
10 OUTPUT 722;"PRESET NORM"          !DCV、NRDGS 1、AUTO、TARM AUTO、TRIG SYN
20 OUTPUT 722;"MEM FIFO"             !読み取り値メモリをオン、FIFOモード
30 OUTPUT 722;"TRlG EXT"             !トリガ・イベント = 外部
40 OUTPUT 722;"EXTOUT BCOMP, NEG"    !バースト完了イベント、立下がりTTL
50 OUTPUT 722;"NRDGS 15, AUTO"       !15回の読み取り/チャネル
55                                   !外部スキャナを構成
60 OUTPUT 709;"SADV EXTIN"           !マルチメータのEXTOUT信号でスキャナを進める
70 OUTPUT 709;"CHCLOSED EXT"         !各チャネル閉鎖後に立下がりパルスを出力
80 OUTPUT 709;"SCAN 201- 206"        !スロット200のスキャナでチャネル01～06をスキャン
85                                   !さらに、チャネル01に進み、スキャンを開始
90 END
```

入力完了

入力完了イベント(ICOMPイベント)は、読み取りごとに1 µsのパルスを生成するPCOMPイベントと似ています。ただし、ICOMPイベントを指定した場合は、A/Dコンバータが入力信号の積分は完了したものの、読み取りが完了していない場合に、パルスが発生します(図20を参照)。ICOMPイベントは、スキャナ・チャネル当たり1回の読み取りを実行する場合に、外部スキャナで使用できます。このイベントは、より低速(リレー式)のスキャナを使用している場合に重要です。ICOMPイベントは、読み取りが完了する前に発生するため、RCOMPイベントの場合よりもスキャナは早く次に進みます。次のプログラムでは、ICOMPイベントを使って、6つのスキャナ・チャネルの1つ1つで1回の読み取りを実行しています。ライン40によってトリガのバッファリングがオンにされていることに注意してください。これによって、スキャナが現在の読み取りが完了する前にチャネル閉鎖パルスを出力しても、マルチメータがTRIGGER TOO FASTエラーを生成することを防ぎます。本例の接続図については、図21を参照してください。

```
10 OUTPUT 722;"PRESET NORM"          !DCV、NRDGS、1、AUTO、TARM AUTO、TRIG SYN
20 OUTPUT 722;"MEM FIFO"             !読み取り値メモリをオン、FIFOモード
30 OUTPUT 722;"TRIG EXT"             !トリガ・イベント = 外部
40 OUTPUT 722;"TBUFF ON"             !トリガ・バッファリングをオン
50 OUTPUT 722 "EXTOUT ICOMP,NEG"     !入力完了EXTOUT、立下がりTTL
55                                   !外部スキャナを構成
60 OUTPUT 709;"SADV EXTIN"           !マルチメータのEXTOUT信号でスキャナを進める
70 OUTPUT 709;"CHCLOSED EXT"         !各チャネル閉鎖後に立下がりパルスを出力
80 OUTPUT 709;"SCAN 201- 206"        !スロット200のスキャナでチャネル01〜06をスキャン
85                                   !さらに、チャネル01に進み、スキャンを開始
90 END
```

アパーチャ波形

指定した場合には、アパーチャ波形イベント(APERイベント)は、A/Dコンバータによる入力信号の測定プロセスを表す波形を出力します。アパーチャ波形は、表示値の測定プロセスを表すだけでなく、オートゼロおよびオートレンジ測定の実行プロセスも表します。この波形を用いることによって、外部スイッチング機器をマルチメータと同期させることができます。例えば、電気的に安定した環境を確保し、確度の高い測定を実現するためには、A/Dコンバータが各読み取り値を積分している間、外部スイッチング機器の動作を中断しなければならない場合もあります。これは、APERイベントをオンにすることによって、またアパーチャ波形からA/Dコンバータが入力信号が積分されていないことが分かる場合にだけ、外部スイッチングが発生するようにプログラミングすることによって行なうことができます。次のプログラム行は、正の(立上がり)極性を持つAPERイベントをオンにします(図20を参照)。

```
OUTPUT 722;"EXTOUT APER,POS"
```

サービス・リクエスト

指定した場合には、サービス・リクエスト・イベント(SRQイベント)は、マルチメータがGPIBサービス・リクエストを生成する度に、1 µsのパルスを生成します。このイベントを用いることによって、外部機器(特に、GPIBに接続できない機器)に、1つまたは複数の指定したイベントが発生し、サービス・リクエストを生成したことを知らせることができます(サービス・リクエストについては、第3章の「ステータス・レジスタの使用方法」を参照してください)。

注記

ステータス・イベントによってレジスタのSRQビットが設定された場合、そのビットはクリアされるまで(例:CSBコマンド)設定されたままになります。指定した場合

には、SRQ(RQSコマンド)をアサートするためにオンにされた何らかのステータス・イベントが発生する度に、EXTOUT SRQパルスが発生します。EXTOUT SRQパルスは、必ずしもSRQビットが設定される度に発生するわけではなく、オン状態にあるステータス・イベントが発生する度に発生します。

次のプログラムでは、SRQイベントを使って、マルチメータを外部機器と同期させています。このプログラムは、サブプログラムをマルチメータにダウンロードします。サブプログラムがコントローラによって呼び出された場合には(ライン20)、10k(Ω)サーミスタを使った高確度の温度測定用にマルチメータが構成されます。サブプログラムが呼び出され、実行されると、ビット0がステータス・レジスタに設定されます(プログラム・メモリ実行完了)。これによって、GPIB SRQ(ライン30によってオン)がアサートされ、**Ext Out**コネクタでパルスが発生します(ライン40によって指定)。このパルスは、マルチメータが構成済みで、測定を実行できる状態にあることを外部機器に信号で知らせます。

```
10 OUTPUT 722;"SUB EXTSRQ"              ! "EXTSRQ"という名前でサブプログラムを保存
20 OUTPUT 722;-"PRESET NORM"            ! PRESET,TRIG SYN、TARM AUTO、NRDGS 1、AUTO
30 OUTPUT 722;"RQS 1"                   !サブプログラム実行完了ビットをオン
40 OUTPUT 722;"EXTOUT SRQ,POS"          !SRQ EXTOUTイベント、立上がりパルス
50 OUTPUT 722;"OHMF 10E3"               !2端子抵抗、10kΩレンジ
60 OUTPUT 722;"NPLC 100"                !100 PLCの積分時間
70 OUTPUT 722;"OCOMP ON"                !オフセット補正をオン
80 OUTPUT 722;"TRIG EXT"                !外部トリガ・イベント
90 OUTPUT 722;"MATH CTHRM10K"           !10kΩサーミスタ演算をオン
100 OUTPUT 722;"CSB"                    !ステータス・レジスタをクリア
110 OUTPUT 722;"SUBEND"                 !サブプログラムの終わり
120 OUTPUT 722;"CALL EXTSRQ"            !サブプログラムを呼び出す
130 END
```

EXTOUT ONCE

EXTOUT ONCEコマンドを実行すると、マルチメータの**Ext Out**コネクタ上に単一の1μsのパルスが発生します。EXTOUT ONCEを実行した後は、モードがOFFに戻ります(EXTOUT信号はオフになります)。次のプログラムに示されているように、EXTOUT ONCEは、サブプログラムの中でサブプログラムまたはサブプログラムのセグメントの終了を外部機器に知らせるのに有用です。

```
10 OUTPUT 722;"SUB EXTONCE"           !サブプログラムを"EXTONCE"という名前で保存
20 OUTPUT 722;"EXTOUT ONCE"           !DC電圧信号への切替えを
25                                    !外部機器に信号で知らせる
30 OUTPUT 722;"PRESET FAST            !高速読み取り、TARM SYN、TRIG AUTO"
40 OUTPUT 722;"MEM FIFO"              !読み取り値メモリをオン、FIFOモード
50 OUTPUT 722;"NRDGS 20"              !20回の読み取り/トリガ
60 OUTPUT 722;"TARM SGL"              !20回の読み取りをトリガ
70 OUTPUT 722;"EXTOUT ONCE"           !抵抗測定への切替えを
75                                    !外部機器に信号で知らせる
80 OUTPUT 722;"OCOMP ON"              !オフセット補正
90OUTPUT 722;"OHM 1E3"                !2端子抵抗、1KΩレンジ
100 OUTPUT 722;"NRDGS 40"             !40回/トリガの読み取り
110 OUTPUT 722;"TARM SGL              !40回の読み取りをトリガ"
120 OUTPUT 722;"SUBEND"               !サブプログラムの終わり
130 OUTPUT 722;"CALL EXTONCE"         !サブプログラムを呼び出す
140END
```

演算

各演算は、各読み取り値に対する特定の演算を実行したり、一連の読み取り値に対してデータを保存します。マルチメータは、読み取り値に対してヌル、スケーリング、パーセント、dB、dBm、フィルタリング、実効値または温度関連の演算を実行することができます。統計処理演算や合否判定演算は、読み取り値を変更するのではなく、読み取り値に付随する情報を保存します。このセクションでは、演算をオン/オフする方法を説明する他、各演算について詳細に説明します。

リアルタイム対後処理

演算は、リアルタイムに行なうことも、後処理で行なうことも可能です。リアルタイム演算をオンにした場合、演算は、読み取り直後に、各読み取り値に対して実行されます。結果は、読み取り値メモリに保存するか、GPIBを経由して出力することができます。後処理演算(STATとPFAILを除く)をオンにした場合、この演算は、各読み取り値が削除されるか、読み取り値メモリからディスプレイまたはGPIB出力バッファにコピーされる時に実行されます。(メモリ内の読み取り値が後処理演算によって変更されることはありません)。STATまたは PFAIL後処理演算は、MMATHコマンドの実行直後に、メモリ内の読み取り値を使って実行されます。統計処理演算については、結果は統計レジスタに保存されます。合否判定演算については、許容範囲外の読み取り値は、ステータス・レジスタのビット番号1を設定し、上限値と下限値のいずれを超えたかによって、FAILED HIGHかFAILED LOWを表示します。

演算のオン

演算をオンにするには、MATHコマンド(リアルタイムの場合)またはMMATHコマンド(後処理の場合)に続いて演算パラメータ(DB、DBM、FILTER、NULL、PERC、PFAIL、RMS、SCALE、STAT、または温度関連のパラメータのうちの1つ)を送信します。温度関連のパラメータのリストについては、このセクションで後述の「温度の測定」を参照してください)。演算をオンにした後は、オフにするか、電源を入れ直すか、RESETを実行するか、PRESETのうちの1つを実行するまで、演算はオン状態のままです。例えば、NULL演算をオンにするには、以下を送信します。

```
OUTPUT 722; "MATH NULL"  !リアルタイムNULL演算をオン
                                または
OUTPUT 722; "MMATH NULL"  !後処理NULL演算オン
```

最高2つまでの演算を同時にオンにできます。演算は、コマンド内の並びに従って、各読み取り値に対して実行されます。例えば、NULL演算とSCALE演算をオンにするには、以下を送信します。

```
OUTPUT 722;"MATH NULL, SCALE"  !リアルタイムNULLおよびSCALEをオン
                                または
OUTPUT 722;"MMATH NULL, SCALE" !後処理NULLおよびSCALEをオン
```

オンになっている演算をすべてオフにするには、以下を送信します。

```
OUTPUT 722;"MATH OFF" !すべてのリアルタイム演算をオフ
                                または
OUTPUT 722;"MMATH OFF" !すべての後処理演算をオフ
```

後で、MATH OFFまたはMMATH OFFコマンドによってオフにした演算を再度オンにすることが可能です。単一の演算を再度オンにするには(2つの演算が以前にオンにされていた場合には、2つの演算のうちの最初の演算だけをオンにします)、以下

を送信します。

```
OUTPUT 722;"MATH CONT" !1つのリアルタイム演算を再度オン
                      または
OUTPUT 722;"MMATH CONT" !1つの後処理演算を再度オン
```

2つの以前にオンにされた演算を再度オンにするには、以下を送信します。

```
OUTPUT 722;"MATH CONT,CONT" !2つのリアルタイム演算を再度オン
または
OUTPUT 722;"MMATH CONT,CONT" !2つの後処理演算を再度オン
```

演算レジスタ

表23には、リアルタイム演算または後処理演算によって使用されるレジスタが示されています。

表23. 演算レジスタ

レジスタ名	レジスタの内容
DEGREE	FILTERおよびRMSの時定数
LOWER	STATSの最小読み取り値
MAX	PFAIL演算の上限値
MEAN	STATSの読み取り値の平均
MIN	PFAILの下限値
NSAMP	STATSのサンプル数
OFFSET	NULLおよびSCALE演算の減数
PERC	PERC演算のパーセント値
REF	DB演算の基準値
RES	DBM演算の基準インピーダンス
SCALE	SCALE演算の除数
SDEV	STATSの標準偏差
UPPER	STATSの最大読み取り値
PFAILNUM	不良が発生する前にPFAILに合格した読み取り値の数

SMATHコマンドを用いることによって、値を任意の演算レジスタ(SDEVを除く)に書き込むことができます。例えば、値22をDEGREEレジスタに入れるには、以下を送信します。

```
OUTPUT 722;"SMATH DEGREE,22"
```

RMATHコマンドを用いることによって、任意のレジスタの値を読み取ることができます。例えば、次のプログラムは、RESレジスタの値を読み取り、出力します。

```
10 OUTPUT 722;"RMATH RES"
20 ENTER 722;A
30 PRINT A
40 END
```

ヌル

NULL演算は、(最初の読み取り値に続く)各読み取り値から値を減算します。式は次のようになります。

結果 = 読み取り値 - OFFSET

ここで、

OFFSETは、OFFSETレジスタに保存されている値です(一般的には、最初の読み取り値)。
読み取り値は、最初の読み取り値に続く任意の読み取り値です。

NULL演算を選択すると、最初に行なわれた読み取り(リアルタイム)または最初にメモリから読み取られた読み取り値(後処理)が、OFFSETレジスタに保存されます。この読み取り値の値は、後続のすべての読み取り値から減算されます。最初の読み取り値をヌル値にしたくない場合には、SMATHコマンドを使って別の値をOFFSETレジスタに書き込むことができます。ただし、最初の読み取りが行なわれるか(リアルタイム)、リコールされる(後処理)のを待ってから、値を変更する必要があります。

NULL演算は、一般に、より正確な2端子抵抗測定を行なうために用いられます。そのためには、2端子抵抗測定を選択し(OHMコマンド)、試験用リード線の両端を短絡させます。それから、NULL演算をオンにします。最初の読み取り値(リード抵抗)がOFFSETレジスタに保存されます。試験用リード線を測定する未知の抵抗に接続します。そうすると、マルチメータは、NULL演算がオフになるまで、後続のすべての読み取り値からOFFSETレジスタの値を減算します。この方法は、4端子抵抗測定ほど正確ではありません。これは、連結された試験用リード線の抵抗が、未知の抵抗に接続された場合と異なることが考えられるためです。また、試験用リード線の抵抗は、一連の測定に対して1回だけしかチェックされないため、試験用リード線の抵抗が変化している可能性があります。

次のプログラムは、20個の読み取り値に対して、リアルタイムでのNULL演算を実行します。NULLコマンドの実行後に、最初の読み取りがライン50によってトリガされます。そうすると、OFFSETレジスタの値が3.05に変わります。20回の読み取りがライン90によってトリガされ、3.05が各読み取り値から減算されます。

```
10OPTION BASE 1                      !配列の番号付けは1から開始
20 DIM Rdgs(20)                      !20個の読み取り値用の配列を宣言
30 OUTPUT 722;"PRESET NORM"          !PRESET、NRDGS 1、AUTO、DCV 10
40 OUTPUT 722;"MATH NULL"            !リアルタイムNULL演算をオン
50 OUTPUT 722;"TRIG SGL"             !1回の読み取りをトリガ、OFFSETに保存
60 OUTPUT 722;"SMATH OFFSET,3.05"    !3.05をOFFSETレジスタに書き込む
70 OUTPUT 722;"NRDGS 20"             !20回の読み取り/トリガ
80 OUTPUT 722;"TRIG SYN"             !SYNトリガ・イベント
90 ENTER 722;Rdgs(*)                 !SYNイベント、NULL補正された読み取り値を入力
100 PRINT Rdgs(*)                    !NULL補正読み取り値を出力
110 END
```

次のプログラムは、20個の読み取り値に対して、後処理NULL演算を実行します。MMATH NULLコマンドの実行後に、21回の読み取りが行なわれ、読み取り値がFIFOモードで読み取り値メモリに保存されます。ライン80は、OFFSETレジスタに保存されている最初の読み取り値をリコールします。そうすると、OFFSETレジスタの値が3.05に変わり、NULL演算が各読み取り値に対して実行されます。

```
10 OPTION BASE 1                     !配列の番号付けは1から開始
20 DIM Rdgs(20)                      !20個の読み取り値用の配列を宣言
30 OUTPUT 722;"PRESET NORM"          !PRESET、NRDGS 1、AUTO、DCV 10
40 OUTPUT 722;"MEM FIFO"             !読み取り値メモリをオン、FIFOモード
50 OUTPUT 722;"MMATH NULL"           !後処理NULL演算をオン
```

```
60 OUTPUT 722;"NRDGS 21"              !21回の読み取り/トリガ
70 OUTPUT 722;"TRIG SGL"              !読み取りをトリガ
80 ENTER 722;A                        !暗黙の読み取りを使って最初の読み取り値をリコール
90 OUTPUT 722;"SMATH OFFSET,3.05"     !3.05をOFFSETレジスタに書き込む
100 ENTER 722;Rdgs(*)                 !暗黙の読み取りを使って読み取り値をリコール、
105                                   !各読み取り値に対してNULL演算を実行
110 PRINT Rdgs(*)                     !NULL補正された読み取り値を出力
120 END
```

スケーリング

SCALE演算は、オフセットを減算し、スケール・ファクタで除算することによって、各読み取り値を修正します。式は次のようになります。

結果 = (読み取り値 – OFFSET)/SCALE

読み取り値は、任意の読み取り値です。
OFFSETは、OFFSETレジスタに保存されている値です(デフォルト = 0。最初の読み取り値がNULL演算の場合のようにOFFSETに保存されないことに注意してください)。
SCALEは、SCALEレジスタに保存されている値です(デフォルト = 1)。

デフォルト値によって読み取り値が変更されることはありません(デフォルト値は0を減算し、1で除算します)。SMATHコマンドを用いることによって、OFFSETレジスタまたはSCALEレジスタの値を変更できます。

次のプログラムでは、リアルタイム・スケール演算を使って、20個の読み取り値をそれぞれ2で除算しています。デフォルト値0はOFFSETレジスタに残されているので、読み取り値のスケーリングが終わるまで、減算は行なわれません。

```
10 OPTION BASE 1                      !配列の番号付けは1から開始
20 DIM Rdgs(20)                       !20個の読み取り値用の配列を宣言
30 OUTPUT 722;"PRESET NORM"           !PRESET、NRDGS 1、AUTO、DCV 10、TRIG SYN
40 OUTPUT 722;"NRDGS 20"              !20回の読み取り/トリガ
50 OUTPUT 722;"MATH SCALE"            !リアルタイムSCALE演算をオン
60 OUTPUT 722;"SMATH SCALE 2"         !2をSCALEレジスタに書き込む
70 ENTER 722;Rdgs(*)                  !SYNイベント、スケーリングされた読み取り値を入力
80 PRINT Rdgs(*)                      !スケーリングされた読み取り値を出力
90 END
```

次のプログラムでは、後処理スケール演算を使って、各読み取り値から値1を減算し、各読み取り値をそれぞれ2で除算しています。

```
10  OPTION BASE 1                     !配列の番号付けは1から開始
20  DIM Rdgs(20)                      !20個の読み取り値用の配列を宣言
30  OUTPUT 722;"PRESET NORM"          !PRESET、NRDGS 1、AUTO、DCV 10、TRIG SYN
40  OUTPUT 722;"MEM FIFO"             !読み取り値メモリをオン、FIFOモード
50  OUTPUT 722;"NRDGS 20"             !20回の読み取り/トリガ
60  OUTPUT 722;"MMATH SCALE"          !後処理SCALE演算をオン
70  OUTPUT 722;"SMATH OFFSET 1"       !1をOFFSETレジスタに書き込む
80  OUTPUT 722;"SMATH SCALE 2"        !2をSCALEレジスタに書き込む
90  OUTPUT 722;"TRIG SGL"             !読み取りをトリガ
100 ENTER 722;Rdgs(*)!                !暗黙の読み取りを使って読み取り値をリコール、
105                                   !各読み取り値に対してSCALE演算を実行
110 PRINT Rdgs(*)                     !演算結果を出力
120 END
```

パーセント

PERC演算は、各読み取り値とPERCレジスタの値の差をパーセントで求めます。式は次のようになります。

結果 = ((読み取り値 – PERC)/PERC) • 100

ここで、
読み取り値は、任意の読み取り値です。
PERCは、PERCレジスタに保存されている値です(電源投入時値 = 1)。

PERC演算を用いることによって、理想値と読み取り値の差(パーセント)を求めることができます。例えば、次のプログラムは、10 VDC電圧測定の%誤差を求めます。ライン60は、PERCレジスタに理想値(10)を入力します。ライン70は、20回の読み取りをトリガします。読み取り値がちょうど10 VDCである場合には、戻り値は次のようになります。

結果 = ((10.1 – 10)/10) • 100 = 0.01 • 100 = 1

```
10 OPTION BASE 1                    !配列の番号付けは1から開始
20 DIM Perc(20)                     !20個の読み取り値用の配列を宣言
30 OUTPUT 722;"PRESET NORM"         !PRESET、NRDGS 1、AUTO、DCV 10、TRIG SYN
40 OUTPUT 722;"NRDGS 20"            !20回の読み取り/トリガ
50 OUTPUT 722;"MATH PERC"           !リアルタイムPERC演算をオン
60 OUTPUT 722;"SMATH PERC 10"       !10をPERCレジスタに書き込む
70 ENTER 722;Perc(*)                !SYNイベント、%差を入力
80 PRINT Perc(*)                    !%差分を出力
90 END
```

次のプログラムは、後処理PERC演算を用いていること以外は、前のプログラムと似ています。

```
10  OPTION BASE 1                   !配列の番号付けは1から開始
20  DIM Perc(20)                    !20個の読み取り値用の配列を宣言
30  OUTPUT 722;"PRESET NORM"        !PRESET、NRDGS 1、AUTO、DCV 10、TRIG SYN
40  OUTPUT 722;"MEM FIFO"           !読み取り値メモリをオン、FIFOモード
50  OUTPUT 722;"NRDGS 20"           !20回の読み取り/トリガ
60  OUTPUT 722;"MMATH PERC"         !後処理PERC演算をオン
70  OUTPUT 722;"SMATH PERC 10"      !10をPERCレジスタに書き込む
80  OUTPUT 722;"TRIG SGL"           !読み取りをトリガ
90  ENTER 722;Perc(*)               !暗黙の読み取りを使って読み取り値をリコール、
95                                  !PERC演算を実行
100 PRINT Perc(*)                   !%差を出力
110 END
```

DB

DB演算は、比をデシベル単位で計算します。式は次のようになります。

結果 = $20 \bullet \log_{10}$(読み取り値/REF)

ここで、

読み取り値は、任意の読み取り値です。
REFは、REFレジスタの値です(デフォルト = 1)。

SMATHコマンドを用いることによって、REFレジスタの値を変更することができます。

次のプログラムでは、リアルタイムDB演算を使って、増幅器の電圧利得を求めています。ライン40は、増幅器の入力電圧(0.1 V)をREFレジスタに保存します。増幅器の出力電圧が測定され、増幅器の利得が計算されます。

```
10 OUTPUT 722;"PRESET NORM"          !PRESET、NRDGS 1、AUTO、DCV 10、TRIG SYN
20 OUTPUT 722;"ACV"                  !AC電圧測定、オートレンジ
30 OUTPUT 722;"SETACV ANA"           !アナログACV法
40 OUTPUT 722;"SMATH REF 0.1"        !0.1をREFレジスタに書き込む
50 OUTPUT 722;"MATH DB"              !リアルタイムDB演算をオン
60 ENTER 722;A                       !SYNイベント、DBを入力
70 PRINT A                           !DBを出力
80 END
```

例えば、入力電圧が0.1 V、出力電圧が10Vの場合には、利得は次のようになります。

$20 \bullet \log_{10}(10/0.1) = 20 \bullet \log_{10}100 = 40\text{dB}$

次のプログラムは、後処理DB演算を用いていること以外は、前のプログラムと似ています。

```
10  OUTPUT 722;"PRESET NORM"         !PRESET、NRDGS 1、AUTO、DCV 10、TRIG SYN
20  OUTPUT 722;"ACV"                 !AC電圧測定、オートレンジ
30  OUTPUT 722;"SETACV ANA"          !アナログACV法
40  OUTPUT 722;"MEM FIFO"            !読み取り値メモリをオン、FIFOモード
50  OUTPUT 722;"SMATH REF 0.1"       !0.1をREFレジスタに書き込む
60  OUTPUT 722;"MMATH DB"            !後処理DB演算をオン
70  OUTPUT 722;"TRIG SGL"            !読み取りをトリガ
80  ENTER 722;A                      !暗黙の読み取りを使って読み取り値をリコール、
85                                   !DB演算を実行
90  PRINT A                          !DB結果を出力
100 END
```

DBM

DBM演算は、1 mWを基準として抵抗に流されたパワーを計算します。式は次のようになります。

結果 = $10 \bullet \log_{10}$(読み取り値^2/RES/1 mW)

ここで、

読み取り値は、任意の電圧読み取り値です。
RESは、RESレジスタの抵抗値です(デフォルト = 50)。

SMATHコマンドを用いることによって、RESレジスタの値を変更することができます。

次のプログラムでは、リアルタイムDBM演算を用いて、スピーカーの入力パワーを求めています。ライン40は、スピーカーのインピーダンスをRESレジスタに保存します(本例の場合、8Ω)。スピーカーの入力電圧が測定され、DBM演算が実行されます。

```
10 OUTPUT 722;"PRESET NORM"          !PRESET、NRDGS 1、AUTO、DCV 10、TRIG SYN
20 OUTPUT 722;"ACV"                  !AC電圧測定、オートレンジ
30 OUTPUT 722;"SETACV ANA"           !アナログACV法
40 OUTPUT 722;"SMATH RES 8"          !8をRESレジスタに書き込む
50 OUTPUT 722;"MATH DBM"             !リアルタイムDBM演算をオン
```

```
60 ENTER 722;A                    !SYNイベント、DBMを入力
70 PRINT A                        !DBMを出力
80 END
```

例えば、入力電圧が10Vの場合、パワーは次のようになります。

$$10\bullet\log_{10}(10^2/8/1\ \mathrm{mW}) = 40.97\mathrm{dBm}$$

次のプログラムは、後処理DBM演算が用いられている以外は、前のプログラムと似ています。

```
10  OUTPUT 722;"PRESET NORM"      !PRESET、NRDGS 1、AUTO、DCV 10、TRIG SYN
20  OUTPUT 722;"ACV"              !AC電圧測定、オートレンジ
30  OUTPUT 722;"SETACV ANA"       !アナログACV法
40  OUTPUT 722;"MEM FIFO"         !読み取り値メモリをオン、FIFOモード
50  OUTPUT 722;"SMATH RES 8"      !8をRESレジスタに書き込む
60  OUTPUT 722;"MMATH DBM"        !後処理DBM演算をオン
70  OUTPUT 722;"TRIG SGL"         !読み取りをトリガ
80  ENTER 722;A                   !暗黙の読み取りを使って読み取り値をリコール、
85                                !DBM演算を実行
90  PRINT A                       !DBM結果を出力
100 END
```

統計処理

STAT演算は、1つのグループの読み取り値に対して5種類の計算を実行し、その結果を5種類の演算レジスタに保存します。5種類の計算とは、標準偏差、平均、サンプル数、最大読み取り値、最小読み取り値です。表24には、STATレジスタとそれらのレジスタの内容が示されています。RMATHコマンドを用いることによって、任意のSTATレジスタを読み取ることができます。

表24. STATレジスタ

レジスタ	保存されている結果
SDEV	標準偏差
MEAN	読み取り値の平均
NSAMP	このグループの測定における読み取り値の数
UPPER	このグループの測定における最大読み取り値
LOWER	このグループの測定における最小読み取り値

次のプログラムでは、リアルタイムSTAT演算を用いて、20個のDC電圧読み取り値に対して、5種類の計算を実行しています。読み取りが行なわれ、読み取り値がコントローラに転送されると、標準偏差が読み取られ、返されます。

```
10  OPTION BASE 1                 !配列の番号付けは1から開始
20  DIM Rdgs(20)                  !20個の読み取り値用の配列を宣言
30  OUTPUT 722;"PRESET NORM"      !PRESET、NRDGS 1、AUTO、DCV 10、TRIG SYN
40  OUTPUT 722"NRDGS 20"          !20回の読み取り/トリガ
50  OUTPUT 722;"MATH STAT"        !リアルタイムSTAT演算をオン
60  ENTER  722 Rdgs(*)            !SYNイベント、読み取り値を入力
70  OUTPUT 722;"RMATH SDEV"       !標準偏差を読み取る
80  ENTER  722;S                  !標準偏差を入力
90  PRINT S                       !標準偏差を出力
100 END
```

次のプログラムは、メモリに保存されている20個の読み取り値に対して、後処理STAT演算を実行します。後処理STAT演算はバッチ演算なので、STAT演算を実行するためにメモリから読み取り値をリコールする必要はありません。また、後処理

STAT演算をオンにする前に、読み取り値を保存する必要があるということにも注意してください(そうしないと、MEMORY ERRORが発生します)。

```
10 OUTPUT 722;"PRESET NORM"            !PRESET、NRDGS 1、AUTO、DCV 10、TRIG SYN
20 OUTPUT 722;"MEM FIFO"               !読み取り値メモリをオン、FIFOモード
30 OUTPUT 722;"NRDGS 20"               !20回の読み取り/トリガ
40 OUTPUT 722;"TRIG SGL"               !読み取りをトリガ
50 OUTPUT 722;"MMATH STAT"             !後処理STAT演算を実行
60 OUTPUT 722;"RMATH SDEV"             !標準偏差を読み取る
70 ENTER  722;S                        !標準偏差を入力
80 PRINT  S                            !標準偏差を出力
90 END
```

合否判定

PFAIL演算は、MAXおよびMINレジスタに設定されているリミット値に照らし合わせて、各読み取り値をテストします。リミットを超えると、ステータス・レジスタのハイ/ロー・ビットが設定されます。また、障害が見つかる前にPFAIL演算に合格した読み取り値の数がPFAILNUMレジスタにログされます。MAXレジスタ、MINレジスタともに、デフォルト値は0です。SMATHコマンドを用いることによって、いずれのレジスタの値も変更することができます。

次のプログラムでは、リアルタイムPFAIL演算を用いることによって、11Vと9Vの上限値と下限値と照らし合わせて、20個のDCV読み取り値がチェックされます。読み取りがトリガされた後に、ステータス・レジスタのHI/LO LIMITビット(ビット2)がチェックされます。1つまたは複数の不良が発生した場合には、PFAILNUMレジスタに問い合せ、不良の内容を返します。

```
10  OPTION BASE 1                      !配列の番号付けは1から開始
20  DIM Rdgs(20)                       !20個の読み取り値用の配列を宣言
30  OUTPUT 722; "PRESET NORM"          !PRESET、NRDGS 1、AUTO、DCV 10、TRIG SYN
40  OUTPUT 722;"MATH PFAIL"            !リアルタイムPFAIL演算をオン
50  OUTPUT 722;"SMATH MIN 9"           !下限値 = 9(V)
60  OUTPUT 722;"SMATH MAX 11"          !上限値 = 11(V)
70  OUTPUT 722;"CSB"                   !ステータス・レジスタをクリア
80  OUTPUT 722;"RQS 2"                 !HI/LOステータス・レジスタ・ビットをオン
90  OUTPUT 722;"NRDGS 20"              !20回の読み取り/トリガ
100 ENTER 722;Rdgs(*)                  !SYNイベント、読み取り値を入力
110 OUTPUT 722; "STB?"                 !ステータス・レジスタの設定ビットを問合せ
120 ENTER 722;A                        !問合せ応答を入力
130 IF BINAND(A,2) THEN                !ビット2が設定されている場合:
140 PRINT "HI/LOW LIMIT TEST FAILED"   !不良メッセージを出力
150 OUTPUT 722; "RMATH PFAILNUM"       !PFAILNUMレジスタに問合せ
160 ENTER 722;B                        !問合せ応答を入力
170 PRINT "NUMBER OF READINGS THAT PASSED BEFORE FAILURE WERE";B
175                                    !PFAILNUM応答を出力
180 ELSE                               !ビット2が設定されていなかった場合:
190 PRINT "HI/LOW LIMIT TEST PASSED"   !試験合格メッセージを出力
200 END IF
210 END
```

次のプログラムは、メモリに保存されている20個の読み取り値に対して、後処理PFAIL演算を用いていること以外は、前のプログラムと似ています。後処理PFAIL演算はバッチ演算なので、PFAIL演算を実行するためにメモリから読み取り値をリコールする必要はありません。また、後処理PFAIL演算をオンにする前に、読み取り値を保存する必要があるということにも注意してください(そうしないと、MEMORY ERRORが発生します)。

```
10  OUTPUT 722;"PRESET NORM"          !PRESET、NRDGS 1、AUTO、DCV 10、TRIG SYN
20  OUTPUT 722;"MEM FIFO"             !読み取り値メモリをオン、FIFOモード
30  OUTPUT 722;"SMATH MIN 9"          !下限値 = 9(V)
40  OUTPUT 722;"SMATH MAX 11"         !上限値 = 11(V)
50  OUTPUT 722;"CSB" "                !ステータス・レジスタをクリア
60  OUTPUT 722;"RQS 2                 !HI/LOステータス・レジスタ・ビットをオン
70  OUTPUT 722;"NRDGS 20"             !20回の読み取り/トリガ
80  OUTPUT 722;"TRIG SGL"             !読み取りをトリガ
90  OUTPUT 722;"MMATH PFAIL"          !後処理PFAIL演算を実行
100 OUTPUT 722;"STB?";                !ステータス・レジスタの設定ビットを問合せ
110 ENTER 722; A                      !問合せ応答を入力
120 IF BINAND(A,2) THEN               !ビット2が設定されている場合:
130 PRINT "Hi/LOW LIMIT TEST FAILED" !不良メッセージを出力
140 OUTPUT 722;"RMATH PFAILNUM"       !PFAILNUMレジスタに問合せ
150 ENTER 722; B                      !問合せ応答を入力
160 PRINT "NUMBER OF READINGS THAT PASSED BEFORE FAILURE WERE";B
165                                   !PFAILNUM応答を出力
170 ELSE                              !ビット2が設定されなかった場合:
180 PRINT "HI/LOW LIMIT TEST PASSED"  !試験合格メッセージを出力
190 END IF
200 END
```

フィルタリング

フィルタ演算は、単極のローパスRCフィルタの出力をシミュレーションします。これによって、長期の傾向を維持しながら、ランダム・ノイズの影響を減少させることができます。式は次のようになります。

結果 = (前の結果)×(DEGREE–1)/DEGREE + 読み取り値/DEGREE

ここで、
前の結果は、最初の読み取り値に初期設定されており、その後は、このFILTER演算の結果に設定されます。
読み取り値は、任意の読み取り値です。
DEGREEは、フィルタのステップ応答を選択します。

DEGREEの値は、ローパス・フィルタのステップ応答に対応します。すなわち、DEGREEの値が20の場合には、最終値の63%を達成するためのステップ応答に、20個の読み取り値が必要です。DEGREEの値を増やすことによって、より低速の応答またはより安定した読み取り値を実現できます。フィルタの実際の時定数(R×C)は、次の式によって求められます。

$$t = \frac{1}{f_s}\left[\frac{1}{ln\frac{DEGREE}{DEGREE-1}} - 1\right]$$

ここで、
t = 時定数(R×C))
f_s = 次のようなサンプリング・レート: l/タイマ間隔(TIMERおよびNRDGSコマンドを使用している場合)またはl/有効間隔(SWEEPコマンドを使用している場合) 。
TIMERまたはSWEEPコマンドを使用していない場合には、本章で前述の「読み取り速度の決定」を参照してください。

DEGREEが10より大きい場合には､次の式によって(R×C)を近似することができます。

$t \approx (l/f_s) \times DEGREE$

例えば(最初の式を用いた場合)、読み取り速度が200Hz、DEGREEが20の場合には、時定数は次のようになります。

$$t = \frac{1}{200}\left[\frac{1}{ln\frac{20}{20-1}} - 1\right] = 0.092 \text{ 秒}$$

同じ読み取り速度とDEGREEで、2番目の式を用いた場合には、次のようになります。

t ≈ (1/200) × 20 = 0.1秒

実効値

RMS演算は、DCV、DSACまたはDSDCコマンドを使ってディジタイズした低周波信号のAC成分とDC成分を合わせた実効値を計算するのに用いることができます。

注記

1 Hz以上の繰り返しAC信号には、RMS演算の代わりに、同期AC測定法を用いることも可能です。AC信号が10Hz以上である場合には、アナログAC法を用いることができます。信号が20Hz以上である場合には、ランダム法を用いることができます。演算をDCV、DSACまたはDSDCコマンドを使ってディジタイズし、STAT演算をオンにすることによって、正弦波のAC成分の実効値を測定することも可能です。多数の読み取りの後には、SDEVレジスタ内の結果は、入力信号のAC成分の実効値になります。

RMS演算は、読み取り値と最初に2乗された前の結果を用いて、続くFILTER演算の平方根を求めます。RMS演算式は次のようになります。

$$\text{結果} = \sqrt{\frac{\text{前の結果}^2 \bullet (DEGREE - 1)}{DEGREE + \frac{\text{読み取り値}^2}{DEGREE}}}$$

ここで、
前の結果は、最初の読み取り値に初期設定されており、その後、このFILTER演算の結果に設定されます。
読み取り値は、最新の読み取り値です。
DEGREEは、フィルタのステップ応答を選択します。

温度の測定

温度関連の演算は、サーミスタまたはRTD(測温抵抗体)の測定抵抗を華氏または摂氏温度の読み取り値に変換します。表25には、温度関連の演算について記述されています。抵抗測定は、2端子抵抗(OHMコマンド)か4端子抵抗(OHMFコマンド)のいずれかで行なうことができます。最高の確度を実現するためには、4端子抵抗モードを使用します。典型的な抵抗測定の確度に影響を及ぼす条件は、温度測定の確度にも影響を及ぼします(第3章の「抵抗」および「校正」を参照してください)。

表25. 温度関連の演算

演算	説明
CTHRM2K	結果=2kΩサーミスタ(40653A)の温度(摂氏)
CTHRM	結果=5kΩサーミスタ(40653B)の温度(摂氏)
CTHRM10K	結果=10kΩサーミスタ(40653C)の温度(摂氏)
FTHRM2K	結果=2kΩサーミスタ(40653A)の温度(華氏)
FTHRM	結果=5kΩサーミスタ(40653B)の温度(華氏)
FTHRM10K	結果=10kΩサーミスタ(40653C)の温度(華氏)
CRTD85	結果= αが0.00385の100Ω RTD(40654Aまたは406548)の温度(摂氏)
CRTD92	結果= αが0.003916の100Ω RTDの温度(摂氏)
FRTD85	結果= αが0.00385の100Ω RTD(40654Aまたは40654B)の温度(華氏)
FRTD92	結果= αが0.003916の100Ω RTDの温度(華氏)

次のプログラムは、10kΩサーミスタを使って温度測定を実行し、その結果を摂氏温度で返します。

```
10 OUTPUT 722;"PRESET NORM"          !マルチメータを初期設定、読み取りを中断
20 OUTPUT 722;"OHMF 10E3"            !4端子抵抗測定を選択、10kΩレンジ
30 OUTPUT 722;"MATH CTHRM10K"        !摂氏変換、10kΩサーミスタ
40 OUTPUT 722;"TRIG SGL"             !読み取りをトリガ
50 ENTER 722;A                       !結果を入力
60 PRINT A                           !結果を出力
70 END
```

第5章 ディジタイジング

第5章 ディジタイジング

概要

ディジタイズとは、連続するアナログ信号を一連の離散サンプル(読み取り値)に変換するプロセスのことを言います。図22は、正弦波をディジタイズした結果を示しています。本章では、信号をディジタイズするための各種方法、サンプリング・レートの重要性、レベル・トリガの使用方法について説明します。

注記 本章の情報の補足として、付録DのProduct Note 3458A-2で、ディジタイズした測定に影響を及ぼすトリガおよびタイムベース誤差について説明しています。

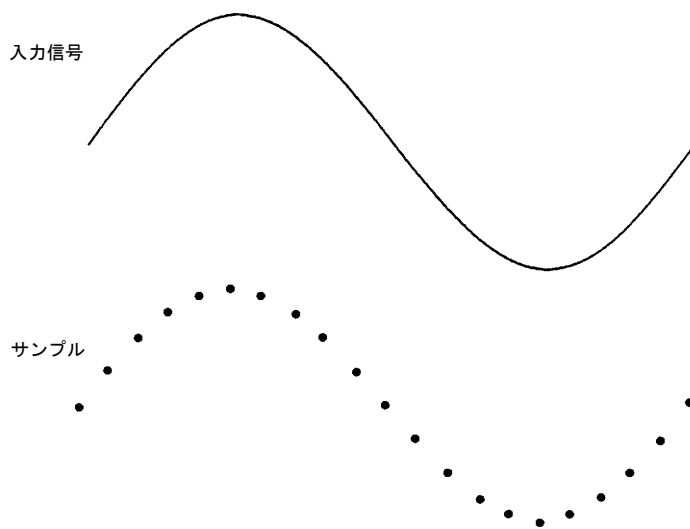



図22. ディジタイズした正弦波

ディジタイズ法

本マルチメータは、直接サンプリングまたはサブ・サンプリングによるDC電圧測定を行なうことによって、信号をディジタイズすることができます。表26には、各ディジタイズ法の特性が要約されています。図23は、各ディジタイズ法のマルチメータの信号経路を単純化したブロック図を示したものです。図24は、すべての種類のディジタイズ法の前面端子の接続を示したものです。

表26. ディジタイズ法

ディジタイズ法	最高サンプリング・レート	帯域幅	必要な繰り返し信号
DCV	100 k/s	DC〜15OkHz[1]	なし
直接サンプリング	50 k/s	DC〜12MHz	なし
サブ・サンプリング	100 M/s[2]	DC〜12MHz	あり

1 レンジによる。詳細については、付録A「仕様」を参照。

2 実効サンプリング・レート(詳細については、本章で後述の「サブ・サンプリング」を参照)。

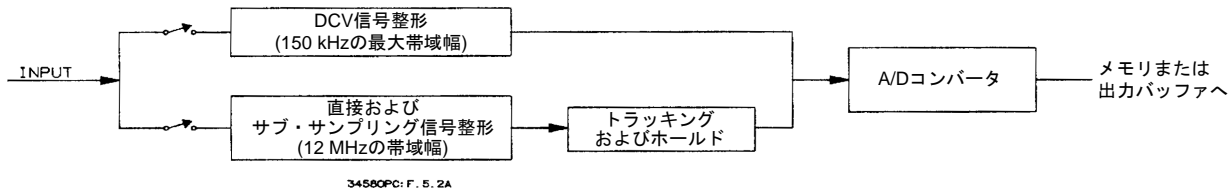



図23. ディジタイズの信号経路

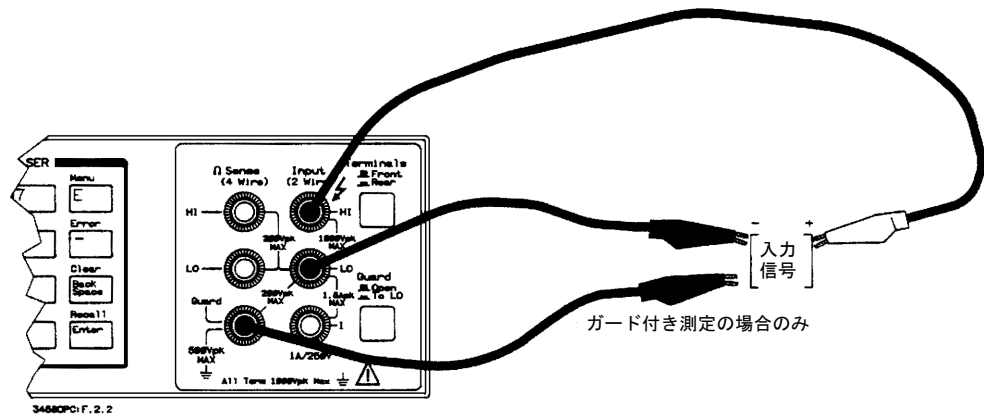



図24. ディジタイズ測定用の接続

ほとんどのディジタイズ・アプリケーションで、サンプリングが開始される度に、本マルチメータは高速モードに入ります。高速モードでは、本器はサンプリング専用になります。つまり、指定した回数のサンプリングが完了するまで、コマンドの処理は一切行ないません。サンプルが高速モードで直接出力バッファに送られた場合、本器は出力バッファから各サンプルが削除されるまで待ってから、次のサンプルを出力バッファに入れます。したがって、バス/コントローラの速度の制限により、サンプルが失われることはありません。(高速モードでない場合には、マルチメータは、新しいサンプルが入手可能になると、出力バッファ内に残存するサンプルを上書きします)。詳細については、第4章の「高速モード」を参照してください。

サンプリング・レート

ナイキスト/サンプリングの定理は、以下を主張します。

> 帯域制限された連続信号にFより高い周波数成分が含まれていない場合、2Fサンプル/秒より速い速度でサンプリングすると、元の信号を歪み(エリアジング)のない状態で復元することが可能です。

本器のサンプリング・レートは、実際には、被測定信号の最高周波数成分の最低でも2倍はなければなりません。サンプリング・レートは、TIMERコマンドによって指定した時間間隔、またはSWEEPコマンドによって指定した*effective_interval*の逆数です。例えば、*effective_interval*を20µsと指定していると仮定します。そうすると、サンプリング・レートは1/20µs = 50,000サンプル/秒になります。

図25は、2Fをわずかに下回るレートでサンプリングした正弦波を示したものです。破線で示されているように、結果は、被測定信号の周波数と大きく異なる**エリアス周波数**になります。

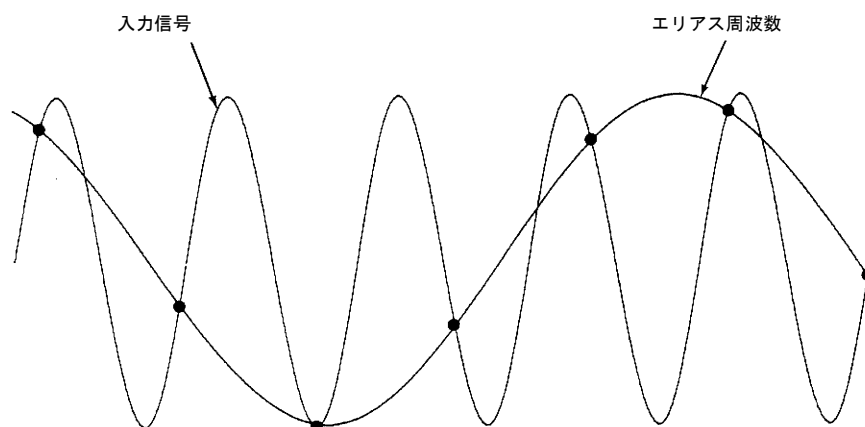



図25. アンダーサンプリングにより生じたエリアジング

ディジタイザの中には、ディジタイザのサンプリング・レートの1/2の急峻なカットオフ周波数を持つエリアジング・ローパス・フィルタが内蔵されているものもあります。このため、入力信号の帯域幅が制限されるため、エリアジングが発生しません。本器はDCVディジタイズ用の可変サンプリング・レートを備えていて、高周波数測定用の上側の帯域を保持するため、アンチ・エリアジング・フィルタは内蔵されていません。エリアジングが懸念される場合には、アンチ・エリアジング・フィルタを外付けしてください。

レベル・トリガ

ディジタイズの際には、信号のゼロ交差が発生した場合や、正または負のピーク振幅の中間点に達した場合など、入力信号上の定義されたポイントでサンプリングを開始することが重要です。レベル・トリガを用いることによって、サンプリングを開始する時期を(電圧やスロープで)指定できます。例えば、図26は、入力信号が立上がりスロープで0V交差した時点でサンプリングが開始されることを示したものです。

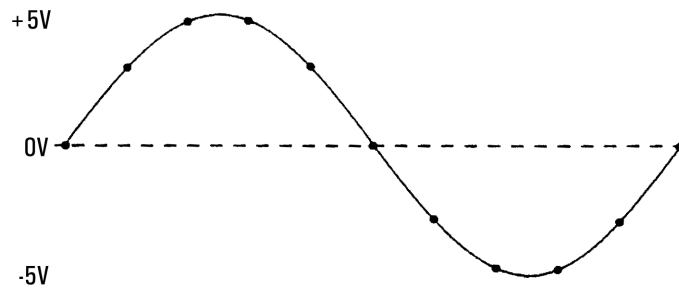



図26. ゼロ交差、立上がりスロープでのレベル・トリガ

レベル・トリガの例

DCVおよび直接サンプリングについては、レベル・トリガをトリガ・イベント(TRIG LEVELコマンド)またはサンプリング・イベント(NRDGS n, LEVELコマンド)として使用できます。サブ・サンプリングについては、レベル・トリガは同期信号源イベントとしてのみ使用可能です(同期信号源については、本章の「サブ・サンプリング」の中で後述します)。このセクションの各プログラム例では、DCV法のディジタイズと10Vレンジを用いています。各種ディジタイズ法でのレベル・トリガの使用方法に関する特定の情報が示された完全なプログラムついては、本章で後述の「DCVディジタイズ」、「直接サンプリング」、「サブ・サンプリング」を参照してください。

LEVELコマンドは、レベル・トリガ電圧を測定レンジのパーセンテージで指定します(レンジについては、本章の各ディジタイズ法の説明の中で後述します)。LEVELコマンドは、レベル検出回路への結合(ACまたはDC)も指定します。

注記

入力信号の結合は、トリガの結合に影響を及ぼす可能性があります。すなわち、入力信号に対してAC結合を選択した場合は(例えば、DSACまたはSSAC)、指定したレベル・トリガの結合に関係なく、レベル・トリガ信号もAC結合されます。ただし、入力信号がDC結合されている場合には(例えば、DCV、DSDC、SSDC)、LEVELコマンドを使って、レベル・トリガ信号の結合を制御できます。レベル・トリガの結合が入力信号の結合に影響を及ぼすことはありません。

SLOPEコマンドは、使用する信号のスロープを指定します。これらのコマンドの電源投入時/デフォルトの値は、現在のレンジの0% のレベル・パーセンテージ(信号がゼロV交差した場合にトリガ)、立上がりスロープ、レベル検出回路へのAC結合を指定します。このため、電源投入時の状態では、LEVELトリガ・イベントを単に指定するだけで(TRIG LEVELコマンド)、図27に示されているようなレベル・トリガを選択できます。

次のプログラムは、レベル・トリガが入力信号が立下がりスロープ(AC結合)の+5V(10Vレンジの50%)に達した場合に発生するように指定しています。図27には、入力信号のピーク値が10Vで、測定レンジが10Vと仮定した場合の結果が示されています。

```
10OUTPUT 722;"PRESET DIG"          !DCVディジタイズ、10Vレンジ
20OUTPUT 722;"TRIG LEVEL"          !レベル・トリガ・イベントを選択
30OUTPUT 722;"SLOPE NEG"           !信号の立下がりスロープでトリガ
40OUTPUT 722;"LEVEL 50, AC"        !10Vレンジの50%でレベル・トリガ、
45                                 !AC結合
50END
```

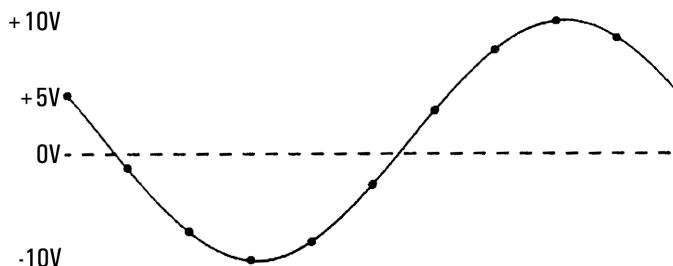



図27. レベル・トリガ、50%、立下がりスロープ、AC結合

次のプログラムは、入力信号が立上がりスロープ(AC結合)で–5V(10Vレンジの–50%)に達した場合に、レベル・トリガが発生するように指定しています。図28には、入力信号のピーク値が±10Vで、測定レンジが10Vと仮定した場合の結果が示されています。

```
10OUTPUT 722;"PRESET DIG"          !DCVディジタイズ、10Vレンジ
20OUTPUT 722;"TRIG LEVEL"          !レベル・トリガ・イベントを選択
30OUTPUT 722;"SLOPE POS"           !信号の立上がりスロープでトリガ
40OUTPUT 722;"LEVEL -50,AC"        !10Vレンジの–50%でレベル・トリガ、
45                                 !(–5V) AC結合
50END
```

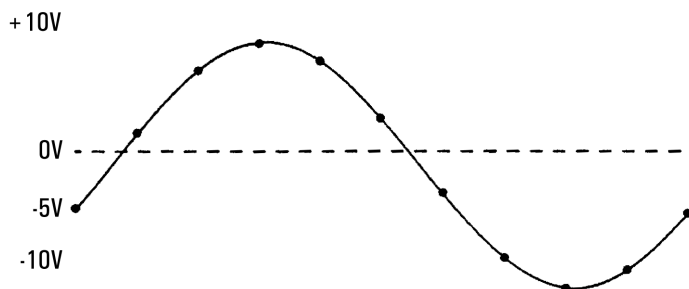



図28. レベル・トリガ、–50%、立上がりスロープ、AC結合

次のプログラムでは、入力信号は、レベル検出回路にDC結合され、–5VのDCレベルに乗った5VピークAC信号です。この場合、レンジの負のパーセンテージ(–25%)を使って、–2.5Vの立上がりスロープでレベル・トリガされます。

結果については、図29を参照してください。

```
10OUTPUT 722;"PRESET DIG"          !DCVディジタイズ、10Vレンジ
20OUTPUT 722;"TRIG LEVEL"          !レベル・トリガ・イベント
30OUTPUT 722;"SLOPE POS"           !信号の立上がりスロープでトリガ
40OUTPUT 722;"LEVEL - 25, DC"      !10Vレンジの–25%でトリガ
45                                 !DC結合
50 END
```

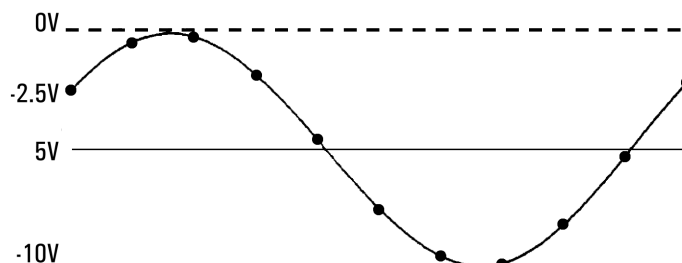



図29. レベル・トリガ、–25%、立上がりスロープ、DC結合

レベル・フィルタ

オンにした場合、レベル・フィルタ機能は、単極のローパス・フィルタ回路をレベル検出回路の入力に接続します。ローパス・フィルタは、75 kHzの3dBポイントを持ち、入力信号の高周波成分が間違ったトリガを引き起こすのを防ぎます。レベル・フィルタをオンにするには、以下を送信します。

```
OUTPUT 722; "LFILTER ON"
```

注記

レベル・フィルタ機能は、周波数/周期測定や、同期(SETACV SYNC)ACVまたはACDCV測定を行なう場合に、マルチメータの高周波ノイズに対する感度を弱めることもできます。

DCVディジタイズ

ディジタイズは、積分時間およびサンプル間隔が短い(ディジタイズする信号の周波数に対する「短さ」) DC電圧測定を指定するだけで実行されます。マルチメータのトラッキング・アンド・ホールド回路は使用されませんが、これはディジタイズと見なされます。DCVディジタイズが直接サンプリング(後述)より優れている点としては、低ノイズ・レベル、高分解能(最高28ビットまで)、100,000サンプル/秒(直接サンプリングの場合、50,000)の最高サンプリング・レートが挙げられます。DCVディジタイズの欠点としては、大きなトリガ・ジッタ(付録A「仕様」を参照)、入力信号のAC結合ができない、150kHzの狭い帯域幅の入力経路(直接サンプリングまたはサブ・サンプリングの場合12MHz)が挙げられます。トラッキング・アンド・ホールド回路はDCVディジタイズには使用されないため、各サンプルの幅ははるかに広くなります(直接サンプリングまたはサブ・サンプリングの場合は2 nsであるのに対して、最低500 ns)。

PRESET DIGコマンドは、50,000サンプル/秒のサンプリング・レートでのDC電圧測定用にマルチメータを構成します。PRESET DIGは、入力信号が立上がりスロープでゼロ交差した場合に、3μsの積分時間とレベル・トリガを選択します。PRESET DIGによって実行される主なコマンドを以下に示します。

`TARM HOLD --`	トリガを中断
`TRIG LEVEL --`	LEVELトリガ・イベント
`LEVEL 0,AC --`	レンジの0%(0V)でレベル・トリガ、AC結合
`TIMER 20E-6 --`	20μsのサンプル間隔
`NRDGS 256,TIMER --`	256サンプル/トリガ、TIMERサンプリング・イベント
`DCV 10 --`	DC電圧測定、10Vレンジ
`DELAY 0 --`	遅延なし
`APER 3E-6 --`	3μsの積分時間
`MFORMAT SINT --`	単精度整数メモリ・フォーマット
`OFORMAT SINT --`	単精度整数出力フォーマット
`AZERO OFF --`	オートゼロ機能をオフ
`DISP OFF --`	ディスプレイをオフ

PRESET DIGを実行した後に、TIMER間隔を短くし、APERコマンドを使って積分時間を短くすることによって、サンプリング・レートを上げることができます。DCVの最低積分時間は500 nsです。

DCVに関する注釈

- DCVディジタイズについては、積分時間が1.4μs以下の場合には、SINTメモリ/出力フォーマットを使用します。積分時間が1.4μsより長い場合には、DINTメモリ/出力フォーマットを使用します。(これらのフォーマットの詳細については、第4章を参照してください)。

注記　最高速度での読み取り値メモリ / コントローラへのサンプルの転送を実現するために、最大10.8μsまでの積分時間については、SINT出力/メモリ・フォーマットを使用することが可能です。ただし、積分時間が1.4μsより長い場合には、A/Dコンバータは、SINTフォーマットが対応できる以上の桁数の分解能を生成します(最下位ビットが切り捨てられます)。10.8μsより長い積分時間でSINT出力/メモリ・フォーマットを使用した場合には、常に、マルチメータは、A/Dコンバータから入って来るデータを変換しなければならないだけでなく、高速モードを維持することはできません。積分時間が10.8μsより長い場合には、DINTメモリ/出力フォーマット(高速モードと互換性のあるフォーマット)を使用してください。

- TIMERサンプリング・イベントまたはSWEEPコマンドを使って測定を行なっている場合には必ず、オートレンジはオフになります。PRESET DIG (10Vレンジ)によって選択したレンジを使用するか、そのレンジをDCVまたはRANGEコマンドの最初のパラメータ(*max_input*パラメータ)として指定することができます。*max_input*パラメータと、それによって選択されるレンジを以下に示します。

max._input パラメータ	選択レンジ	フルスケール
0〜0.12	100mV	120mV
>0.12〜1.2	1V	1.2V
>l.2〜12	10V	12V
>12〜120	100V	120V
>120〜1E3	1000V	1050V

- マルチメータのトリガ階層(トリガ・アーム・イベント、トリガ・イベント、サンプリング・イベント)は、DCVディジタイズに適用されます。DCVディジタイズ用のトリガ階層の詳細については、第4章を参照してください。DCVディジタイズには、TIMERサンプリング･イベントとNRDGS *n*,TIMERコマンドまたはSWEEPコマンドのいずれかを使用できます。NRDGSコマンドとSWEEPコマンドは互換性があるので、マルチメータは、どちらか最後に指定されたコマンドを使用します。(SWEEPコマンドを使用した場合には、サンプリング・イベントは自動的にTIMERに設定されます)。

- アパーチャ時間とは、マルチメータが入力信号を実際にサンプリングしている時間のことを言います。トラッキング・アンド・ホールドを使った直接サンプリングやサブ・サンプリングの場合、アパーチャ時間はA/Dコンバータの積分時間と等しく、500ns〜1sの範囲で変更可能です。マルチメータは、アパーチャ時間内で、入力信号を実効的にアベレージングします。アパーチャ時間内で信号が変化すると、振幅誤差が生じます。表27は、選択したアパーチャ時間内で3dBの振幅誤差が発生する入力信号周波数と、これらのアパーチャ時間内で実現される分解能のビット数を示したものです。

表27. 振幅誤差と分解能対アパーチャ

アパーチャ時間	分解能のビット数	3dBの誤差が発生する場合の周波数
2ns	16	100MHz
500ns	15	400kHz
1µs	16	206kHz
3µs	17	69kHz
6µs	18	35kHz
100µs	21	2kHz

DCVの例

次のプログラムは、100,000サンプル/秒の速度で256個のDC電圧サンプルを取り出し、SINTフォーマットを使って読み取り値メモリに保存します。取り出されたサンプルは、SINT出力フォーマットを使ってコントローラに転送されます。コントローラは、サンプルをSINTフォーマットから変換し、保存します。ライン100を削除することによって、読み取り値メモリを使わずに、直接コントローラにサンプルが転送されます。ただし、コントローラとGPIBが、少なくとも200kバイト/sの速度でサンプルを転送できなければなりません。転送できな場合は、TRIGGER TOO FASTエラーが生成されます。詳細については第4章の「GPIBによる高速転送」を参照してください。

```
10OPTION BASE 1                              !配列の番号付けは1から開始
20Num_samples=256                            !サンプル数を指定
30INTEGER Int_samp(l:256) BUFFER             !整数バッファを作成
40ALLOCATE REAL Samp(l:Num_samples)          !サンプル用の実数配列を作成
50ASSIGN @Dvm TO 722                         !マルチメータのアドレスを割り当て
60ASSIGN @Int_samp TO BUFFER Int_samp(*)     !I/Oパス名をバッファに割り当て
70OUTPUT @Dvm;"PRESET DIG"                   !TARM HOLD、DCV、10Vレンジ、トリガ当たり
71!256サンプル、TIMERサンプリング・イベント、TIMER間隔 = 20µs、トリガ・
75!レベル(0%、AC結合)、3µsの積分時間、SINTフォーマット
80OUTPUT @Dvm;"TIMER 10E-6"                  !10µsのサンプル間隔
90OUTPUT @Dvm;"APER 1.4E-6"                  !100kHzのサンプリング・レートの場合の最大アパーチャ
100OUTPUT @Dvm;"MEM FIFO"                    !読み取り値メモリをオン、FIFOモード
110OUTPUT @Dvm; "TARM SYN"                   !同期トリガ・アーム・イベント
120TRANSFER @Dvm TO @Int_samp;WAIT           !SYNイベント、読み取り値を読み取り値メモリに転送
121!さらに、コンピュータの整数配列に入れる。
122!コンピュータの整数フォーマットはSINTと同じなので、
123!ここではデータ変換は不要(整数配列は必要)
130OUTPUT @Dvm; "ISCALE?"                    !SINTフォーマットのスケール・ファクタを問合せ
140ENTER @Dvm;S                              !スケール・ファクタを入力
150FOR I=1 TO Num_samples
160  Samp(I)=Int_samp(I)                     !各整数読み取り値を実数フォーマットに変換
165  !(次行で整数のオーバーフローの発生を防ぐために必要)
170  R=ABS(Samp(I))                          !絶対値を使ってOVLDをチェック
180  IF R>=32767 THEN PRINT "OVLD"            !OVLDの場合、過負荷メッセージを出力
190  Samp(I)=Samp(I)*S                       !読み取り値にスケール・ファクタを乗算
200  Samp(I)=DROUND(Samp(I),4)               !4桁に丸める
210NEXT I
220END
```

直接サンプリング

直接サンプリングは、サンプルがリアルタイムで取得され、連続する各サンプルは前のサンプルから指定された時間間隔で隣接しているという点で、DCVディジタイズと似ています。両者の違いは、直接サンプリングでは、マルチメータのトラッキング・アンド・ホールド回路が使用されること、入力経路が広帯域幅(12 MHz帯域幅)であることです。さらに、直接サンプリングの場合は、DCVディジタイズに比べてトリガ・ジッタは少ないものの、測定ノイズは大きくなります(付録A「仕様」を参照)。

トラッキング・アンド・ホールド回路は、入力信号のサンプルを高速で取得し、A/Dコンバータがそのサンプルを積分している間値を保持します。トラッキング・アンド・ホールド回路を用いることによって、各サンプルの幅が最低500 ns(DCVの場合)から2 ns(直接サンプリングの場合)に減少します。このため、直接サンプリングは、急峻なパルスのピーク振幅を捕捉するといったアプリケーションなどに適しています。直接サンプリングの欠点は、最高サンプリング・レートが50,000サンプル/秒と、DC電圧の場合の100,000サンプル/秒に比べて低速であることです。

直接サンプリングは、DSACまたはDSDCコマンドを使って指定します。DSACコマンドは、AC結合を選択します。これによって、入力信号のAC成分だけが測定されます。DSDCコマンドは、DC結合を選択します。これによって、入力信号のAC成分とDC成分が合わせて測定されます。

図30は、正弦波の入力に対して直接サンプリングを使って取り出された20個のサン

プルを示したものです(番号は、サンプルが取り出された順番を示します)。直接サンプリングを用いた場合、サンプル間の可能な最小間隔は20μsです。

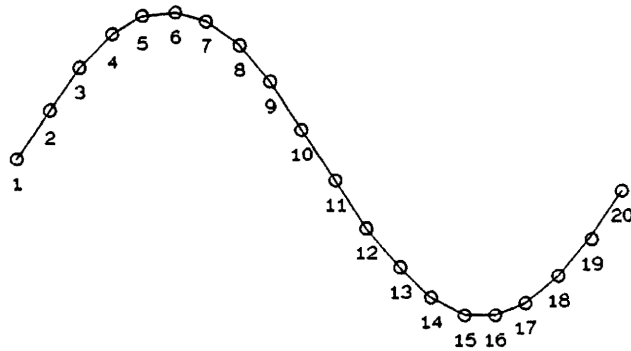



図30. 直接サンプリング

直接サンプリングに関する注釈

- 直接サンプリング測定には、オートレンジは使用できません。したがって、DSACまたはDSDCコマンドの最初のパラメータ(*max._input*パラメータ)としてレンジを指定しなければなりません。*max._input*パラメータとそれによって選択されるレンジは以下の通りです。

max._input パラメータ	選択されるレンジ	フルスケール	
		SINTフォーマット	DINTフォーマット
0〜0.012	10mV	12mV	50mv
>0.012〜0.120	100mV	120mV	500mV
>0.120〜1.2	1V	1.2V	5.0V
>1.2〜12	10V	12V	50V
>12 〜120	100V	120V	500V
>120〜1E3	1000V	1050V	1050V

DINTメモリ/出力フォーマットを使用している場合は、直接サンプリングのフルスケール値は10mV、100mV、1V、10Vおよび100Vのレンジの500%(5倍)になることに注意してください。これは特に、レベル・トリガのパーセンテージを指定する場合に考慮に入れることが大切です。レベル・トリガ電圧を指定する場合は、レンジのパーセンテージを使用します。例えば、入力信号のピーク値が20Vで、10Vレンジを使用していると仮定します。15Vでのレベル・トリガを希望する場合は、150%のレベル・トリガのパーセンテージを指定します(LEVEL 150コマンド)。(周波数が2MHz以上、振幅がレンジの120%以上の信号を測定している場合は、マルチメータの増幅器のスルーレートを超えてしまう可能性があります。最高12MHzまでの周波数レンジの120%以下の信号であれば、スルーレート・エラーが生じることはありません)。

- マルチメータのトリガ階層(トリガ・アーム・イベント、トリガ・イベント、サンプリング・イベント)は、直接サンプリングに適用されます。つまり、これらのイベントは、直接サンプリングが開始される前に、適切な順番で発生します。トリガ階層の詳細については、第4章を参照してください。直接サンプリングには、TIMERサンプリング・イベントとNRDGS *n*,TIMERコマンドまたはSWEEPコマンドのいずれかを使用できます(SWEEPコマンドの方がプログラムが簡単です)。NRDGSコマンドとSWEEPコマンドは互換性があるので、マルチメータは、どち

らか最後に指定したコマンドを使用します。(SWEEPコマンドを使用した場合は、サンプリング・イベントは自動的にTIMERに設定されます)。

- 1 MHzの周波数成分を持つ入力信号を直接サンプリングした場合は、補間器のセトリング時間のために、最初のサンプルに誤差が生じる恐れがあります。最初のサンプルを正確なものにするためには、最初のサンプルの前に500nsの遅延を挿入します(DELAY 500E-9コマンド)。

直接サンプリングの例

次のプログラムは、DC結合で直接サンプリングしたディジタイズの一例です。SWEEPコマンドは、30µsの間隔と、200個のサンプルを指定しています。レベル・トリガは、10Vレンジの250%に設定されます(10Vの250% = 25V)。サンプルは、DINTフォーマットで読み取り値メモリに送られ、さらにコントローラに送られ、変換され、出力されます。ライン110を削除することによって、読み取り値メモリを使わずに、直接コントローラにサンプルが転送されます。ただし、コントローラとGPIBが、少なくとも134kバイト/sの速度でサンプルを転送できなければなりません。できない場合は、TRIGGER TOO FASTエラーが生成されます。詳細については第4章の「GPIBによる高速転送」を参照してください。

```
10OPTION BASE 1                                  !配列の番号付けは1から開始
20INTEGER Num_samples,I,J,K                      !整数変数を作成
30Num_samples = 200                              !200個のサンプル
35ASSIGN @Dvm TO 722                             !マルチメータのアドレスを指定
40ASSIGN @Buffer TO BUFFER [4*Num_samples]       !サンプル用のコントローラ・バッファを設定、
45!(4バイト/サンプル * 200サンプル = 800バイト)
50ALLOCATE REAL Samp(l:Num_samples)              !サンプル用の実数配列を作成
60OUTPUT @Dvm;"PRESET FAST"                      !DINTフォーマット、TARM SYN、TRIG AUTO
70OUTPUT @Dvm; "SWEEP 30E - 6,200"               !30µsの間隔、200個のサンプル
80OUTPUT @Dvm; "DSDC 10"                         !直接サンプリング、10Vのレンジ
90OUTPUT @Dvm;"LEVEL 250, DC"                    !レンジ(25V)の250%でのレベル・トリガ
100OUTPUT @Dvm;TRIG LEVEL"                       !レベル・トリガ・イベント
110OUTPUT @Dvm; "MEM FIFO"                       !読み取り値メモリをオン、FIFOモード
120TRANSFER @Dvm TO @Buffer;WAIT                 !コントローラへサンプルを転送
130OUTPUT @Dvm; "ISCALE?"                        !DINTフォーマットのスケール・ファクタを問合せ
140ENTER @Dvm;S                                  !スケール・ファクタを入力
150FOR I=1 TO Num_samples
160  ENTER @Buffer USING "#,W,W";J,K             !1個の16ビットの2の補数ワードを
161  !各変数JとKに入力(# = 文ターミネータは不要;
165  !W = 16ビットの2の補数としてデータを入力)
170  Samp(I)=(J*65536.+K+65536.*(K<0))           !実数に変換
180  R=ABS(Samp(I))                              !絶対値を使ってOVLDをチェック
190  IF R>2147483647 THEN PRINT "OVLD"            !過負荷が発生した場合は、メッセージを出力
200  Samp(I)=Samp(I)*S                           !スケール・ファクタを適用
210  Samp(I)=DROUND(Samp(I),8)                   !変換された読み取り値を丸める
220  PRINT Samp(I)                               !読み取り値を出力
230NEXT I
240END
```

サブ・サンプリング

サブ・サンプリング(シーケンシャル・サンプリングとも言われる)では、マルチメータは、入力信号の各周期で、1個または複数のサンプルを取得します。連続する各周期で、開始サンプリング・ポイントを遅らせて、より多くのサンプルが取得され

ます。多数の周期が発生し、指定された数のサンプルが取得された後、サンプルを復元して、入力信号の周期と同じ周期を持つコンポジット波形を復元することが可能です。

サブ・サンプリングの利点は、最小間隔がDCVディジタイズの場合が10µs、直接サンプリングの場合が20µsであるのに対して、10nsの最小間隔で実効的にサンプルを取得できるという点にあります。つまり、サブ・サンプリングを用いることによって、最高12 MHz(サブ・サンプリングの信号経路の上側の帯域)までの周波数成分を持つ信号をディジタイズすることが可能です。サブ・サンプリング測定では、アパーチャが2nsのトラッキング・アンド・ホールド回路が用いられます。サブ・サンプリング(および直接サンプリング)の場合は、DCVディジタイズに比べてトリガ・ジッタが少なくなります(付録A「仕様」を参照)。サブ・サンプリングの欠点は、入力信号が周期的(繰り返し)でなければならず、リアルタイム測定ではないという点です。

サブ・サンプリングは、SSACまたはSSDCコマンドを使って指定します。SSACコマンドは、入力信号のAC成分だけをディジタイズするAC結合サブ・サンプリングを選択します。SSDCコマンドは、信号のAC成分とDC成分を合わせてディジタイズするDC結合サンプリングを選択します。

サブ・サンプリングの原理

サブ・サンプリングでは、コンポジット波形のサンプルは非常に密接に隣接しています。つまり、コンポジット波形のサンプル間の間隔(*effective_interval*)は、DCV法や直接サンプリング法よりもはるかに狭く(はるかに高速の実効サンプリング・レートに)設定できます。例えば、サンプル間の*effective_interval*が5µsの10kHzの繰り返し入力信号をディジタイズしなければならないと仮定します。これは、1/5e-6、すなわち1秒当たり200,000サンプルのサンプリング・レートに相当します(DCVまたは直接サンプリングを用いた場合、最高サンプリング・レートはそれぞれ1秒当たり100,000サンプルと50,0000サンプルであるため、このアプリケーションでは不可能です)。図31は、サブ・サンプリングを用いることによって、これをどのように実現できるかを図示したものです。*effective_interval*は5µsと指定されています。また、指定したサンプル数は20です。*effective_interval*と合計サンプル数は、SWEEPコマンドによって指定します。*effective_interval*とサンプル数を指定すると、マルチメータは、必要なバースト(サンプルのグループ)の数と、各バーストに含めるサンプルの数を計算します。

本例の場合、入力信号の最初の周期で、マルチメータは5個のサンプルから成るバーストを取得します。2番目の周期では、トリガ・ポイントを5µs遅延させ、5個のサンプルから成る別のバーストを取得します。残りの2つの周期ではそれぞれ、トリガ・ポイントをさらに5µs遅延させ、5個のサンプルから成るバーストを取得します。図32に示されているように、すべてのサンプルが正しい順序で並べられ、5µsの間隔で隣接する20個のサンプルから成る入力信号の1周期となります。本例の場合、**実効サンプリング・レート**は1秒当たり200,000サンプルになります。

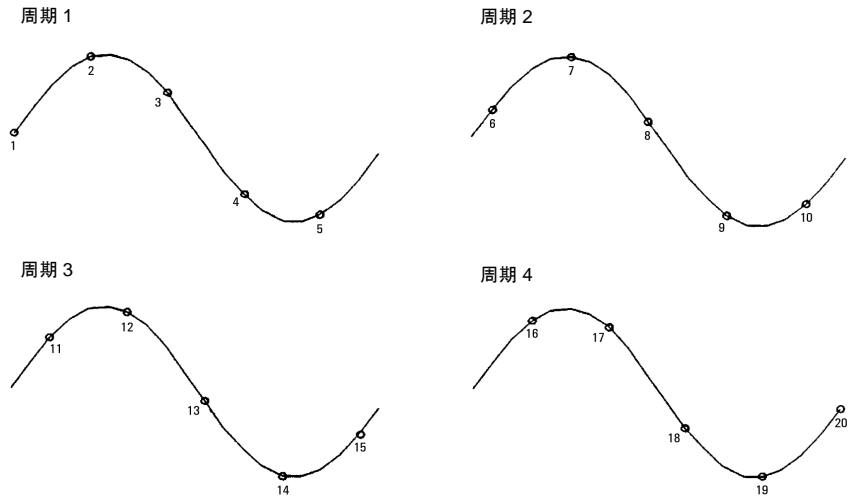



図31. サブ・サンプリングの例

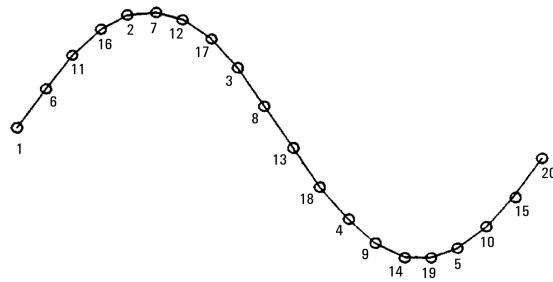



図32. コンポジット波形

同期信号源イベント

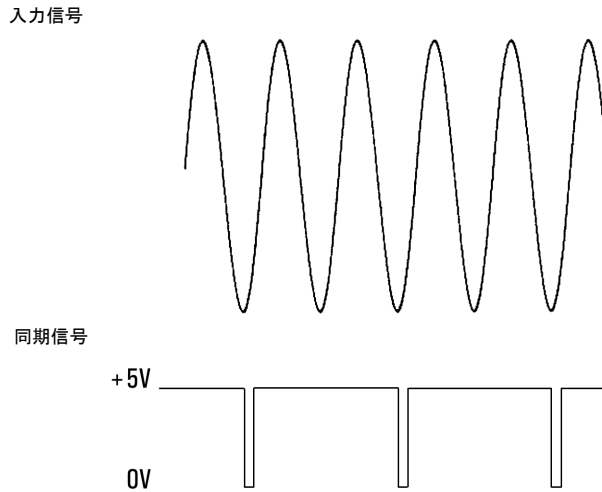
前のサブ・サンプリングの例では、マルチメータが入力波形の周期に合わせて何らかの形で自己同期化できるものと仮定されていました。これが同期信号源イベントの役割です。EXTイベントかLEVELイベントを同期信号源イベントとして使用できます。EXT同期信号源イベント(SSRC EXTコマンドによって指定)は、マルチメータの**Ext Trig**コネクタの立下がりエッジで発生します。このため、入力信号と同期した外部パルスが必要です。図33は、典型的な入力信号と、必要な同期信号を示したものです。図33では、同期信号は必ずしも入力信号のすべての周期に対して1回ずつ発生する必要はないことに注意してください。ただし、入力信号と時間的に同期する必要があります。

同期信号

図33. EXT同期信号源の典型的な同期信号

LEVEL同期信号源イベント(電源投入時/デフォルトの同期信号源イベント)は、入力信号が指定したスロープで指定した電圧レベルに達すると発生します(レベル・トリガ)。図31は、LEVEL同期信号源の動作を示したものです(本例の場合、LEVELは0%、立上がりスロープ、AC結合と指定されています)。最初の同期信号源イベントは、入力信号が立上がりスロープで0V交差した場合に発生します。マルチメータは、サンプル・バースト(この場合5個のサンプル)を取得します。次に信号源イベントが発生すると(入力信号の周期2)、マルチメータはトリガ・ポイントを遅延させ、さらに5個のサンプルを取得します。このプロセスは、指定した回数のサンプリングが完了するまで繰り返されます。

次の例では、SSDCコマンドを使って、5VのDCレベルに乗った5Vのピーク値を持つ1 MHzの信号をディジタイズします。SWEEPコマンドは、10nsの*effective_interval*で、4000個のサンプルを取得するように、マルチメータに指示します。ライン60〜80は、LEVEL同期信号源イベントの電圧レベルとスロープをプログラムします。これによって、入力信号の最初の周期が7.5 VDC(10Vレンジの75%)に達したときに、サンプリングが開始します。ライン90は、本質的に同期信号源イベントをオンにするトリガ・アーム・イベントの条件を満たします。

```
10OUTPUT 722;"PRESET FAST"          !TARM SYN、TRIG AUTO、DINTフォーマット
20OUTPUT 722;"MEM FIFO"             !読み取り値メモリをオン、FIFOモード
30OUTPUT 722;"MFORMAT SINT"         !SINT読み取り値メモリ・フォーマット
40OUTPUT 722."SSDC 10"              !サブ・サンプリング、10V レンジ、レベル同期信号源
45 !EVENT (DEFAULT EVENT)
50OUTPUT 722;"SWEEP l0E-9,4000"     !4000個のサンプル、10nsの実効間隔
60OUTPUT 722;"LEVEL 75 DC"          !レンジの75%でレベル・トリガ、DC結合
70OUTPUT 722;"SLOPE POS"            !立上がりスロープでレベル・トリガ
80OUTPUT 722;"SSRC LEVEL"           !LEVEL同期信号源イベント
90OUTPUT 722; "TARM SGL"            !サンプリングをオン
100END
```

サブ・サンプリングに関する注釈

- サブ・サンプリングの場合、トリガ・イベントとサンプリング・イベントの要件は無視されます(これらのイベントについては、第4章を参照してください)。サブ・サンプリングに適用されるトリガ・イベントは、トリガ・アーム・イベント(TARMコマンド)と同期信号源イベント(SSRCコマンド)だけです。

- サブ・サンプリングには、NRDGSコマンドは使用できません。サンプル数と*effective _interval*を指定するには、SWEEPコマンドを使用しなければなりません。サブ・サンプリングの最小*effective_interval*は10nsです。最高サンプリング・レートは50000サンプル/秒です(サンプル間20µs)。

- サブ・サンプリング測定には、オートレンジは使用できません。したがって、SSACまたはSSDCコマンドの最初のパラメータ(*max._input*パラメータ)としてレンジを指定しなければなりません。*max._input*パラメータと、それによって選択されるレンジは以下の通りです。

max._ input パラメータ	選択されるレンジ	フルスケール
0～0.012	10mV	12mV
>0.012～0.120	100mV	120mV
>0.120～1.2	1V	1.2V
>1.2～12	10V	12V
>12～120	100V	120V
>120～ lE3	1000V	1050V

直接サンプリングの場合と同じように、最高でレンジの500%までのレベル・トリガ電圧を指定できます。ただし、SINTフォーマットでは、レンジの120%を超えるサンプルを処理することはできません。

- SSACまたはSSDCコマンドを実行した時に読み取り値メモリがオフの場合には、出力フォーマットが自動的にSINTに設定されます(メモリ・フォーマットは変更されません)。後で、別の測定ファンクションに変更した場合には、出力フォーマットは前に指定されていたフォーマットに戻ります。サブ・サンプリングやGPIBへのサンプルの直接出力には、SINT出力フォーマットを使用しなければなりませんが､サンプルが読み取り値メモリの最初に配置されている場合は(次の注釈を参照)、いずれの出力フォーマットでも使用可能です。そのためには、SSACまたはSSDCコマンドを実行する前に、読み取り値メモリをオンにする必要があります(読み取り値メモリがオンになっている場合には､SSACまたはSSDCコマンドを実行しても出力フォーマットは変わりません)。

- 読み取り値メモリをオンにした状態でサブ・サンプリングを行なう場合には、読み取り値メモリがFIFOモードで､しかも空でなければなりません(MEM FIFOを実行すると、読み取り値メモリがクリアされます)。さらに、トリガ・アーム・イベントが発生する前に、メモリ・フォーマットがSINTになっていなければなりません。そうなっていない場合は、トリガ・アーム・イベントが発生して、1つのサンプルも取得されない場合に、SETTINGS CONFLICTエラーが生成されます。

- 1MHz の周波数成分を持つ入力信号をサブ・サンプリングした場合は、補間器のセトリング時間のために、最初のサンプルに誤差が生じる恐れがあります。最初のサンプルを正確なものにするためには、DELAY 500E-9コマンドを使って500nsの遅延を挿入します。(サブ・サンプリングの場合、遅延は同期信号源イベントと各バーストの最初のサンプルの間に挿入されます。サブ・サンプリングのデフォルト遅延は0秒です)。

サンプルのメモリへの送信

サンプルが読み取り値メモリに直接送信された場合は(MEM FIFOコマンド)、マルチメータは、コンポジット波形を生成するサンプルを自動的に順序付けし直します。例えば、次のプログラムでは、サブ・サンプリングされたデータは、必要なSINTメモリ・フォーマットを使って読み取り値メモリに送信されます。マルチメータは、サンプルを正しい順番でメモリに入れます。サンプルは次に、DREAL出力フォーマットを使ってコントローラに転送されます(サブ・サンプリングされたデータを読み取り値メモリに最初に配置した場合には、SINT出力フォーマットの使用に制限があります)。

```
10OPTION BASE 1                     !配列の番号付けは1から開始
20REAL Samp(l:200) BUFFER           !バッファ配列を作成
30ASSIGN @Dvm TO 722                !マルチメータのアドレスを割り当て
40ASSIGN @Samp TO BUFFER Samp(*)    !バッファを割り当て
50OUTPUT @Dvm;"PRESET FAST"         !TARM SYN、TRIG AUTO、DINTフォーマット
60OUTPUT @Dvm;"MEM FIFO"            !先入れ先出し(FIFO)読み取り値メモリ
70OUTPUT @Dvm;"MFORMAT SINT"        !SINTメモリ・フォーマット
80OUTPUT @Dvm;"OFORMAT DREAL"       !倍精度実数出力フォーマット
90OUTPUT @Dvm;"SSDC 10"             !サブ・サンプリング、10Vレンジ、DC結合
100OUTPUT @Dvm;"SWEEP 5E-6,200"     !5µsの実効間隔、200個のサンプル
110TRANSFER @Dvm TO @Samp;WAIT      !コントローラのバッファにサンプルを転送
120FOR I=1 TO 200
130  IF ABS(Samp(I))=lE+38 THEN     !過負荷を検出
140    PRINT "Overload Occurred"    !過負荷メッセージを出力
150  ELSE                           !過負荷が発生しなかった場合:
160    Samp(I)=DROUND(Samp(I),5)    !5桁に丸める
170    PRINT Samp(I)                !各サンプルを出力
180  END IF
190NEXT I
200END
```

コントローラへのサンプルの送信

サンプルがコントローラに直接送信される場合は、アルゴリズムを用いて、サンプルを順序付けし直して、コンポジット波形を生成する必要があります。SSPARM?コマンドは、アルゴリズムの3つのパラメータを返します。返される最初のパラメータは、N個のサンプルから成る測定バーストの個数です。返される2番目のパラメータは、Nの値です。例えば、10kHzの信号をサブ・サンプリングしようとして、*effective_interval*が5µsの22個のサンプルを指定します。本例では、マルチメータは、それぞれ6個のサンプルから成るバーストを2個取得します。各バーストは、前のバーストから5µs遅延されています。SSPARM?の戻り値は、2、2および6です。

サブ・サンプリングの場合、最高サンプリング・レートは、指定した*effective_interval*に関係なく、50000サンプル/秒です。(20µs以上の*effective_interval*を指定した場合、マルチメータはもはやサブ・サンプリングは行なわず、直接サンプリングを行ないます)。サンプルを直接コントローラに送信する場合には(サンプル当たり2バイトのSINTフォーマットを使用)、GPIB/コントローラは、1秒当たり100kバイトの最高速度でデータを処理できなければなりません。できなければ、TRIGGER TOO FASTエラーが生成されます。

次ページのプログラムでは、5Vのピーク値を持つ10 kHzの信号がディジタイズするのに、SSACコマンドが用いられています。SWEEPコマンドは、2µsの*effective_interval*で(Eff_int変数)、1000個のサンプルを取得するようにマルチメータに指示します(Num_samples変数)。この測定では、同期信号源イベントのデフォルトのレベル・ト

リガ(0%、AC結合、立上がりスロープの入力信号からのトリガ)が用いられます。ライン110は、SYNイベントを生成し、サンプルを直接コンピュータに転送します。ライン230～400は、サブ・サンプリングされたデータをソートして、コンポジット波形を生成します。コンポジット波形は、Wave_form配列に保存されます。

```
10OPTION BASE 1                        !配列の番号付けは1から開始
20INTEGER Num_samples,Inc,I,J,K,L      !変数を宣言
30Num_samples=1000                     !サンプル数を指定
40Eff_int=2.0E-6                       !実効間隔を指定
50INTEGER Int_samp(1:1000) BUFFER      !整数バッファを作成
60ALLOCATE REAL Wave_form(l:Num_samples)!保存データ用の配列を作成
70ALLOCATE REAL Samp(l:Num_samples)    !サンプル用の配列を作成
80ASSIGN @Dvm TO 722                   !マルチメータのアドレスを割り当て
90ASSIGN @Int_samp TO BUFFER Int_samp(*)!バッファ I/Oパス名を割り当て
100OUTPUT @Dvm;"PRESET FAST;LEVEL;SLOPE;SSDC 10;SWEEP ";Eff_int,Num_samples
101!高速演算、TARM SYN、デフォルトのレベルとスロープ、サブ・サンプリング(SINT
105!出力フォーマット)、10Vのレンジ、2μsの実効間隔、1000個のサンプル
110TRANSFER @Dvm TO @Int_samp;WAIT     !SYNイベント、読み取り値を整数配列に転送;
111!コンピュータの整数フォーマットはSINTと同じなので、
115!ここではデータ変換は不要(整数配列は必要)
120OUTPUT @Dvm; "I SCALE?"             !SINTフォーマットのスケール・ファクタを問合せ
130ENTER @Dvm;S                        !スケール・ファクタを入力
140OUTPUT @Dvm; "SSPARM?"              !サブ・サンプリング・パラメータを問合せ
150ENTER @Dvm;Nl,N2,N3                 !サブ・サンプリング・パラメータを入力
160FOR I=1 TO Num_samples
170  Samp(I)=Int_samp(I)               !各整数読み取り値を実数フォーマットに変換
175  !(次行で整数のオーバーフローの発生を防ぐために必要)
180  R=ABS(Samp(I))                    !絶対値を使ってOVLDをチェック
190  IF R>=32767 THEN PRINT "OVLD"     !OVLDの場合、過負荷メッセージを出力
200  Samp(I)=Samp(I)*S                 !読み取り値にスケール・ファクタを乗算
210  Samp(I)=DROUND(Samp(I),4)         !4桁に丸める
220NEXT I
225 !-------------------サンプルをソート----------------------
230Inc=Nl+N2                           !合計バースト数
240K=l
250FOR I=1 TO N1
260  L=I
270  FOR J=l TO N3
280    Wave_form(L)=Samp(K)
290    K=K+1
300    L=L+Inc
310  NEXT J
320NEXT I
330FOR I=Nl+l TO Nl+N2
340  L=I
350  FOR J=1 TO N3-1
360    Wave_form(L)=Samp(K)
370    K=K+1
380    L=L+Inc
390  NEXT J
400NEXT I
410END
```

サンプリングしたデータの表示

次ページのプログラムは、コントローラのCRTにディジタイズしたデータをプロットします(この特定のプログラムはサブ・サンプリングを使用しているため、サブルーチン*Plot_it*が実際のプロットを実行することはありません)。このプログラムは、捕捉中のデータを表示できるので、ディジタイズ・プログラムを開発する場合(特に、サブ・サンプリングする場合)に有用です。このプログラムは、サンプル間にベクトルを描くだけなので(リニア補間)、サンプリング・レートが被測定信号の周波数の10倍以上である場合に有効です。サンプリング・レートが入力信号の周波数の10倍以下である場合には、このプログラムは誤った表示の入力信号をプロットします。図34は、このプログラムによって生成される典型的なプロットを示したものです。

注記 Agilent 3458オプション005波形解析ライブラリは、ディジタイズしたデータを捕捉し、処理するように設計されたソフトウェア・パッケージで、システムの初期化、データの捕捉、データの比較、データのパラメータの計算、データのフーリエ変換の実行、データのプロット/出力を行なう各種ルーチンが含まれています。詳細については、最寄のAgilent営業所にお問い合わせください。

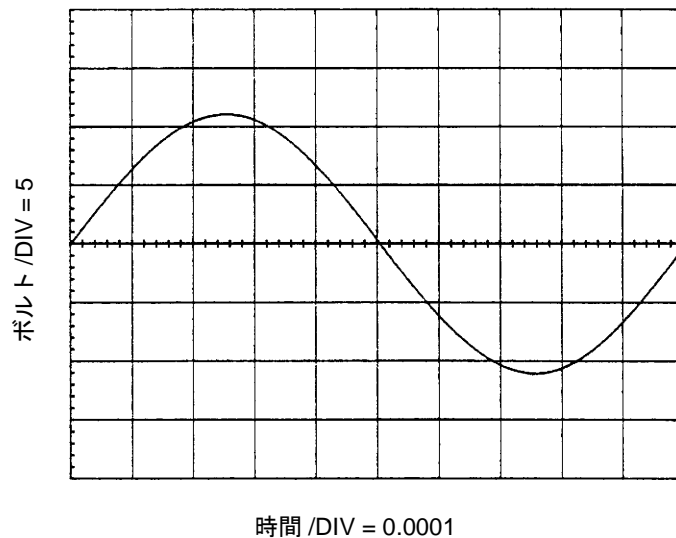



図34. 典型的なプロット波形

```
10OPTION BASE 1                            !配列の番号付けは1から開始
20INTEGER Num_samples,Inc,I,J,K,L          !変数を宣言
30INTEGER Int_samp(l:l000) BUFFER          !整数バッファを作成
40ALLOCATE REAL Wave_form(l:Num_samples)!保存データ用の配列を作成
50ALLOCATE REAL Samp(l:Num_samples)        !サンプル用の配列を作成
60Num_samples=1000                         !サンプル数を指定
70Eff_int=2.0E-6                           !実効間隔を指定
80ASSIGN @Dvm TO 722                       !マルチメータのアドレスを割り当て
90ASSIGN @Int_samp TO BUFFER Int_samp(*)!I/Oパス名をバッファに割り当て
100OUTPUT @Dvm;"PRESET FAST;SSDC 10;SWEEP ";Eff_int, Num_samples
```

```
101!高速演算、TARM SYN、サブ・サンプリング(SINT出力フォーマット)、10Vのレンジ
102!2µsの実効間隔、1000個のサンプル
110TRANSFER @Dvm TO @Int_samp;WAIT      !SYNイベント、読み取り値を転送
120OUTPUT @Dvm;"ISCALE?"               !SINTフォーマットのスケール・ファクタを問合せ
130ENTER @Dvm;S                        !スケール・ファクタを入力
140OUTPUT @Dvm;"SSPARM?"               !サブ・サンプリング・パラメータを問合せ
150ENTER @Dvm;Nl,N2,N3                 !サブ・サンプリング・パラメータを入力
160FOR I=1 TO Num_samples
170  Samp(I)=Int_samp(I)               !各整数読み取り値を実数フォーマットに変換
175                         !(次行で整数のオーバーフローの発生を防ぐために必要)
180  R=ABS(Samp(I))                    !絶対値を使ってOVLDをチェック
190  IF R>=32767 THEN PRINT "OVLD"      !OVLDの場合、過負荷メッセージを出力
200  Samp(I)=Samp(I)*S                 !読み取り値にスケール・ファクタを乗算
210  Samp(I)=DROUND(Samp(I),4)         !4桁に丸める
220NEXT I
230Inc=Nl+N2                           !Inc = 合計バッファ数
240K=l                                 !サンプルをソート
250FOR I=1 TO Nl                       !     "
260  L=I                               !     "
270  FOR J=l TO N3                     !     "
280    Wave_form(L)=Samp(K)            !     "
290    K=K+l                           !     "
300    L=L+Inc                         !     "
310  NEXT J                            !     "
320NEXT I                              !     "
330FOR I=Nl+l TO Nl+N2                 !     "
340  L=I                               !     "
350  FOR J=l TO N3-1                   !     "
360    Wave_form(L)=Samp(K)            !     "
370    K=K+l                           !     "
380    L=L+Inc                         !     "
390  NEXT J                            !     "
400NEXT I                              !     "
410DISP                                !コントローラのCRTをクリア
420Time_div=l.0E-5                     !プロットの時間/Div
430Volts_div=5                         !プロットの電圧/Div
440Plot_it(Time_div,Volts_div,Wave_form(*),Eff_int)
450END
460SUB Plot_it(Time_div,Volts_div,Wave_form(*),Time_base)
470DIM X_axis$[80],Y_axis$[80]
480GINIT
490GRAPHICS ON
500RAD
510MOVE 35,10
520LDIR 0
530X_axis$="TIME/DIV = "&VAL$(Time_div)
540LABEL X_axis$
550MOVE 15,35
560LDIR PI/2
570Y_axis$="VOLTS/DIV = "&VAL$(Volts_div)
580LABEL Y_axis$
590VIEWPORT 20,110,20,90
600WINDOW 0,10*Time_div,-4*Volts_div,4*Volts_div
610AXES Time_div/5,Volts_div/5,0,0,1,1,1
620GRID Time_div,Volts_div
630Wave x=0
640MOVE Wave_x,Wave_form(BASE(Wav_form,l))
650FOR Wave_y=BASE(Wave_form,l)+l TO SIZE(Wave_form,l)-l+BASE (Wave_form,l)
660  Wave_x=Wave_x+Time_base
```

```
670  DRAW Wave_x,Wave_form(Wave_y)
680NEXT Wave_y
690IF Wave_x>l0*Time_div THEN DISP "More samples taken than displayed"
700SUBEND
```

第6章　コマンド・リファレンス

第6章 コマンド・リファレンス

概要

本章の最初の部分では、マルチメータの言語について説明します。ここでは、コア・コマンド、コマンド・ターミネータ、パラメータ、問合せコマンド、ファンクション・グループごとのコマンド・リスト、およびコマンドを測定ファンクションに関連付けた表が記載してあります。本章の残りの部分では、各コマンドを詳細に(コマンドをアルファベット順で)説明しています。

本章を使用する前に、使う必要のあるマルチメータ機能を前のチュートリアルの章(第2章、第3章、第4章および第5章)で読んでおいてください。これらのチュートリアルの章では、個々のマルチメータ機能を説明しているので、使う必要のあるコマンドがわかります。その後で、本章を使用して、個々のコマンドの詳細を確認してください。本章のコマンドは、以下の形式で説明してあります。

言語規約

マルチメータは、GPIBバス[1] を使用してシステム・コントローラと通信します。GPIBに接続されている個々の測定器は固有のアドレスを持っています。本書で使用している例は、BASIC言語を使用するHewlett-Packardシリーズ200または300コンピュータ用のものです。これらの例では、GPIBセレクト・コードの7とデバイス・アドレスの22(工場アドレス設定)を想定しており、その結果として組み合わせたGPIBアドレスは722となります。プログラミングを簡単にするため、このアドレスは変えないことをお勧めします。

コマンド・ターミネータ

キャリッジ・リターン(*cr*)、改行(*lf*)、セミコロン(;)または最終文字と同時に送られるEOIが、マルチメータへのメッセージの終わり(コマンド・ターミネータ)を示します。コマンドをシステム・コントローラから標準フォーマット(例えば、OUTPUT 722;"TEST")で送ると、通常コントローラによってコマンドの終わりに*cr lf*が追加されます。マルチメータは入力バッファがオフのため(オフが電源投入時の入力モードです)、*cr*は直ちに処理されますが、*lf*はコマンドの実行が完了するまで処理されません。これは、*lf*のためにバスがホールドされ、マルチメータがコマンドの実行を完了するまで(または、GPIB CLEARコマンドが実行されて、コマンドの実行が中断されるまで)、コントローラを再度使用できないということです。バスのホールドは、コマンドを送る際に*cr lf*を抑制するか、入力バッファをオンに(INBUF ONコマンド)することで防げます。次のプログラム行は、#およびKイメージ指示子を使用してマルチメータ・コマンドを送る際に*cr lf*を抑制する方法を示します。

```
OUTPUT 722 USING "#,K";"TEST;"
```

注記 #およびKイメージ指示子は、HP Series 200/300コンピュータに対応しています。コンピュータによりどのように*cr lf*が抑制されるかについては、お使いのコンピュータの操作マニュアルを参照してください。TESTコマンドに続くセミコロンは、マルチメータに対してコマンドの終わりを示し、*cr lf*を抑制する場合にはこれが必要です。

複数のコマンド

複数のコマンドをセミコロンで区切ると、1つのコマンド文字列として送ることができます。例えば、次のコマンド文字列には、3つのマルチメータ・コマンドが含まれています。

```
OUTPUT 722; "TRIG HOLD;DCV 3;NPLC 10"
```

パラメータ

コマンド・パラメータとして指定する数値のフォーマットには、整数、浮動小数点または指数があります。浮動小数点フォーマットのパラメータは、コマンドが整数を必要とする場合には最も近い整数に丸められます。例えば、"SUB 2.49"は**切り下げて**"SUB 2"に丸められ、"SUB 2.5"は**切り上げて**"SUB 3"に丸められます。

パラメータのデフォルト設定

パラメータは、省略するか –1(マイナス1)に置き換えることでデフォルトに設定できます。例えば、最初のパラメータとして10を指定し、2つめのパラメータはデフォルトとする場合には、次のコマンドを送ります。

```
OUTPUT 722;"ACV 10"
```

1. GPIBは、IEEE標準488-1978およびANSI MC1.1.のAgilent Technologiesによるインプリメンテーションです。

または

```
OUTPUT 722;"ACV 10,-1"
```

また、リモートからのみ、2つのカンマを使用してデフォルト値を示すことができます。例えば、最初のパラメータとして10を指定し、2つめのパラメータをデフォルトとする場合には、次のコマンドを送ります。

```
OUTPUT 722;"ACV 10,,"
```

最初のパラメータ(この例では、オートレンジの選択)をデフォルトとし、2つめのパラメータとして0.01を指定するには、次のコマンドを送ります。

```
OUTPUT 722; "ACV,,.01"
```

問合せコマンド

問合せコマンドは疑問符で終わり、ある質問に対して1つまたは複数の応答を返します。例えば、ID?問合せコマンドは、応答*HP 3458A*を返します。

標準問合せコマンド

次の標準問合せコマンドを本章で個別に説明しています。

AUXERR?	LINE?
CALNUM?	MCOUNT?
ERR?	OPT?
ERRSTR?	REV?
ID?	SSPARM?
ISCALE?	STB?
	TEMP?

その他の問合せコマンド

標準問合せコマンドの他にも、マルチメータの構成またはプログラムに使用できる任意のコマンドの末尾に疑問符を付ければ問合せコマンドを作成できます(この種の問合せコマンドは、本章では個別に説明せず、親コマンドと組み合わせてあります。つまり、AZEROコマンドのページであれば、AZEROとAZERO?について記載してあります)。例として、AZEROコマンドは、オートゼロ機能をオンまたはオフにします。可能なオートゼロ・モードは、OFF、ONまたはONCEです。次に示すプログラムのように、疑問符をAZEROコマンドの末尾に付ければ、現在のオートゼロ・モードを判別することができます。

```
10 OUTPUT 722;"AZERO?"
20 ENTER 722;A$
30 PRINT A$
40 END
```

電源投入時のステートでは、マルチメータは問合せコマンドに対して数値応答を返します。例えば、上のプログラムは、数値問合せでONパラメータに対応する1を返します。本章では、数値問合せの等価値を各アプリケーション・コマンドの下にリストしてあります。

パラメータを選択できるコマンド(AZEROコマンドなど)では、コマンドの問合せバージョンでは、現在指定されているパラメータ(または、数値問合せでこのパラメータと等価の値)が返されます。多数のコマンドが、パラメータの選択肢ではなく、秒、V、Ωなどで指定された実際の値を使用します。例えば、APERコマンドは、積分時間を秒で指定します。このコマンドの値の範囲は500ns～1sです。APER?問合せコマンドを送ると、マルチメータは現在指定されている積分時間の実際の値で応答します。

QFORMAT(問合せフォーマット)コマンドを使用すると、問合せ応答が(上に示したように)数値、英字または英数字のいずれになるかを指定できます。例えば、次のプログラムは問合せフォーマットをALPHA(英字)に変更します。これにより、次のプログラムに示すように、マルチメータが英字のコマンド・ヘッダと英字の応答(可能な場合)を返します。

```
10 OUTPUT 722;"QFORMAT ALPHA"
20 OUTPUT 722;"AZERO?"
30 ENTER 722;A$
40 PRINT A$
50 END
```

代表的な応答: AZERO ON

IALPHA問合せフォーマットでは、実際の値を使用するコマンドは英字コマンド・ヘッダと数値応答を返します。例えば、APER?問合せコマンドに対する代表的な応答は次のとおりです。

```
APER 166.667E-03
```

多数の問合せコマンドが、英字応答と数値応答の両方を返すことができます。例えば、NRDGS?問合せコマンドは2つの応答を返します。1番目の応答は数値で、トリガ・イベントあたりの読み取り回数を示します。2番目の応答は英字で(QFORMAT = ALPHAと仮定)、指定したサンプル・イベントを示します。次のプログラムは、NRDGS?問合せコマンドを実行して、応答を出力します。

```
10 OUTPUT 722;"NRDGS?"
20 ENTER 722;A$,B$
30 PRINT A$,B$
40 END
```

問合せコマンドに対する応答は、指定した出力フォーマットに関係なく、常にGPIBを経由したASCII出力フォーマットです。問合せ応答の後、出力フォーマットは前に指定されていたフォーマット(SINT、DINT、SREAL、DREALまたはASCII)に戻ります。

ファンクション・グループ別コマンド

次に示すのは、マルチメータにより認識されるすべてのコマンドをファンクション別(測定ファンクション、デジタイズ、A/Dコンバータなど)に分類したものです。

測定ファンクション
ACDCI
ACDCV
ACI
ACV
DCI
DCV
DSAC
DSDC
FREQ
FUNC
OHM
OHMF
PER
SSAC
SSDC

測定関連
ACBAND
ARANGE
AZERO
DELAY
FIXEDZ
FSOURCE
LFILTER
OCOMP
PRESET (DIG、FASTまたはNORM)
RANGEまたはR
RATIO
SETACV
SSPARM?
TERM

デジタイズ
DSAC
DSDC
LEVEL
LFILTER
SLOPE
NRDGS
PRESET (DIGおよび FAST)
SSAC
SSDC
SSPARM?
SSRC
SWEEP
TIMER

トリガ
EXTOUT
LEVEL
LFILTER
NRDGS
SLOPE
SSRC
SWEEP
TARM
TBUFF
TIMER
TRIGまたは T

読み取り値メモリ
MCOUNT?
MEM
MFORMAT
MSIZE
RMEM

プログラム・メモリ
CALL
COMPRESS
CONT
DELSUB
PAUSE
SCRATCH
SUB
SUBEND

ステート・メモリ
PURGE
RSTATE
SCRATCH
SSTATE

A/Dコンバータ
APER
LFREQ
LINE?
NPLC
RES

ステータス
CSB
RQS
SRQ
STB?

入力/出力
END
INBUF
ISCALE?
OFORMAT
QFORMAT

エラー
AUXERR?
EMASK
ERR?
ERRSTR?

演算
MATH
MMATH
RMATH
SMATH

キーボード
DEFKEY
LOCK
MENU

Bus
ADDRESS
ID?
SRQ

システム
BEEP
DEFEAT
EXTOUT
OPT?
PRESET (DIG、FASTまたはNORM)
QFORMAT
RESET
TONE

ディスプレイ
DISP
NDIG

校正/テスト
ACAL
CAL
CAL?
CALNUM?
CALSTR
REV?
SCAL
SECURE
TEMP?
TEST

GPIBコマンド
ABORT IO
CLEAR
LOCAL
LOCAL LOCKOUT
REMOTE
SPOLL
TRIGGER

コマンドと測定ファンクションとの対応

表28は、特定のファンクションにのみ適用されるマルチメータ・コマンドを示します。黒丸(●)は、そのコマンドが制限なしで適用されることを示します。数字(1〜5)は、そのコマンドが制限付きで適用されることを示します(表の下にある番号付き脚注を参照してください)。空白は、そのコマンドが測定ファンクションには適用されないことを示します。表28に記載されていない他のマルチメータ・コマンドは、制限なしですべての測定ファンクションに適用されます。

表28. コマンドと測定ファンクションの対応

	DCV	DCI	OHM OHMF	ACV ACDCV (ANA)	ACV ACDCV (SYNC)	ACV ACDCV (RNDM)	ACI ACDCI	FREQ PER	DSAC DSDC	SSAC SSDC
ACBAND				•	•	•	•	•		
APER	•	•	•	•			•			
ARANGE[1]	•	•	•	•	•	•	•	•		
AZERO	•	•	•							
FIXEDZ	•		•							
FSOURCE								•		
ISCALE?	•	•	•	•	•	•	•	1	•	•
LEVEL	•				2			3	•	•
LFILTER	•				•			•	•	•
LFREQ	•	•	•	•			•			
(M) MATH[1]	•	•	•	•	•	•	•	•	•	4
MFORMAT	•	•	•	•	•	•	•	1	•	5
NPLC	•	•	•	•			•			
OCOMP			•							
OFORMAT	•	•	•	•	•	•	•	1	•	5
RATIO	•			•	•	•				
SETACV				•	•	•				
SLOPE	•				2			3	•	•
SSPARM?										•
SSRC					•					•
SWEEP	•	•	•	•			•		•	•
TIMER	•	•	•	•			•		•	

1. リアルタイムまたは後処理演算がオンの場合(STATまたはPFAILを除く)またはオートレンジがオンの場合は、SINTまたはDINT出力/メモリ・フォーマットはFREQまたはPER測定には使用しないでください。
2. レベル・トリガは、同期ACVまたはACDCVのデフォルト同期信号源イベントです。ただし、レベル・トリガ電圧とスロープは自動的に決定され、指定することはできません。
3. LEVELトリガ・イベントまたはLEVELサンプル・イベントを、FREQ測定またはPER測定に使用することはできません。ただし、レベル検出回路が周波数または周期測定に使用する電圧レベルまたはスロープは指定することができます。
4. MATHをサブサンプリングに使用することはできません。MMATHはサブサンプリングに使用できます。
5. サブサンプリングでは、読み取り値メモリを使用する場合はメモリフォーマットが SINT でなければなりません。読み取り値メモリを使用しない場合は、出力フォーマットがSINTでなければなりません。

ACAL

自動校正。マルチメータに対して、自己校正の1つまたはすべてを実行するように命令します。

構文 **ACAL [*type*][,*security_code*]**

type *type*パラメータの選択肢は次のとおりです。

type パラメータ	数値 問合せの 等価値	説明
ALL	0	DCV、OHMSおよびAC自動校正を実行します。
DCV	1	DC電圧利得およびオフセット(1番目の備考を参照)
AC	2	ACVフラットネス、利得およびオフセット (2番目の備考を参照)
OHMS	4	OHMS利得およびオフセット(3番目の利得を参照)

電源投入時の*type* = なし
デフォルトの*type* = ALL

security_code 自動校正にセキュリティがかかっている場合は、自動校正を実行するには正しいセキュリティ・コードを入力する必要があります。自動校正にセキュリティがかかっていない場合には、セキュリティ・コードは不要です。セキュリティ・コードと、自動校正のセキュリティ保護とセキュリティ解除の詳細についてはSECUREコマンドを参照してください。

備考

- DCV自動校正はすべての測定機能に適用されるため、ACまたはOHMS自動校正を実行する前に実行してください。ACAL ALLを指定すると、DCV自動校正が他の自動校正の前に実行されます。
- マルチメータは熱的に安定な環境で最低限2時間パワーオンにした後で自動校正する必要があります。最高の確度を得るためには、24時間に1回か、直近の外部校正または直近の自動校正を行ってからマルチメータの温度が±1°C変化した場合にACAL ALLを実行する必要があります。
- AC自動校正では、ACVまたはACDCV(すべての測定法)、ACIまたはACDCI、DSAC、DSDC、SSAC、SSDC、FREQ、PERの測定が向上します。
- OHMS自動校正は、2端子または4端子抵抗測定、DCIおよびACI測定が向上します。
- 自動校正を実行する前に、必ずAC入力信号を切断してください。入力信号をマルチメータに接続したままにすると、自動校正に悪影響を及ぼす場合があります。
- 自動校正の定数は、不揮発性メモリ内に保存されます(この定数は、電源を切断しても元の状態のままです)。電源を入れ直しただけで、自動校正を実行する必要はありません。

- 各自動校正ルーチンの実行に必要な時間は次のとおりです。

```
ALL                    :  11分
DCV                    :  1分
AC                     :  1分
OHMS                   :  10分
```

- **関連コマンド**: CAL、SCAL、SECURE

例

```
OUTPUT 722;"ACAL ALL,3458"   !工場セキュリティ・コードを使用して
                             !すべての自動校正を実行します。
```

ACBAND

AC帯域幅。すべてのACまたはAC+DC測定の入力信号の周波数成分(帯域幅)を指定します。帯域幅を指定することで、マルチメータによる最高速測定のための構成が可能になります。

構文 **ACBAND [*low_frequency*][,*high_frequency*]**

low_frequency 入力信号の予想最低周波数成分を指定します。

電源投入時の*low_frequency* = 20 Hz

デフォルトの *low_frequency* = 20 Hz

high_frequency 入力信号の予想最高周波数成分を指定します。

電源投入時の*high_frequency* = 20MHz

デフォルトの*high_frequency* = 2MHz

備考

- 入力信号の帯域幅に基づく確度と測定速度の仕様については、付録Aを参照してください。
- 同期ACVまたはACDCV(SETACV SYNC コマンド)の場合、帯域幅パラメータがマルチメータによるタイムアウト値とサンプリング・パラメータの計算に使用されます。レベル・トリガ(デフォルト・モード)を使用している場合、入力信号が読み取り中に取り除かれ、時間リミット内に返されないと、読み取りが完了するように測定法はランダム法に変更されます(読み取り後、測定法はSYNCに戻ります)。同期ACVまたはACDCVの場合、指定した帯域幅が測定対象信号の信号成分に対応していることが非常に重要です。
- オートレンジをオンにした状態での周波数または周期測定では、帯域幅パラメータがオートレンジングに必要な時間量の算出に使用されます。これらの測定の場合、指定した帯域幅(特に*low_frequency*)が、測定対象信号の周波数成分に対応していることが非常に重要です。
- 入力信号の周波数成分が分からない場合には、ACBANDパラメータをデフォルトにしてください。

- **問合せコマンド**　ACBAND?問合せコマンドは、カンマで区切られた2つの数字を返します。1番目の数字は現在指定されている*low_frequency*で、2つめの数字は*high_frequency*です。詳細については、本章の始めにある「問合せコマンド」を参照してください。

- **関連コマンド**: ACDCI、ACDCV、ACI、ACV、FREQ、FUNC、PER、SETACV

例
```
OUTPUT 722;"ACBAND 500,1000" !入力信号を500～1000Hzに
                             !指定します。
```

ACDCI、ACDCV、ACI、ACV

FUNCコマンドを参照してください。

ADDRESS

マルチメータのGPIBアドレスを設定します(フロントパネルからのみ)。このアドレスは不揮発性メモリに保存され、電源が遮断されても失われません。

構文 **ADDRESS *value***

value *value*パラメータは、0～31の整数です。

電源投入時の*value* = 前に保存されたアドレス(工場設定 = 22)
デフォルトの*value* = なし、パラメータが必要

備考

- アドレス31を設定すると、マルチメータのアドレスは実際には変わらず、マルチメータがTalk Onlyモードに設定されます。このモードでは、バス上のコントローラなしで、マルチメータが読み取り値をGPIBプリンタへ直接出力します(ASCII出力フォーマットを使用する必要があります)。Talk Onlyモードでは、マルチメータの**TALK**インジケータが点灯します。バス上のコントローラを使用してアドレス31を指定することはできません。マルチメータをTalk Onlyモードから外すには、**Reset**キーを押すか、31以外のアドレスを指定します。

- コントローラのアドレスは通常21です。コントローラのアドレスは、GPIBバス上の他のデバイス用に使用しないでください。

- マルチメータは、CMOS RAM不良(補助エラービット12)を検出すると、アドレスを22に設定します。

- ADDRESS?問合せ。マルチメータの前面パネルから、**Address**キー(シフト+**Local**キー)を使用して現在のアドレスを読み取ることができます。

- **関連コマンド**: ID?

APER

アパーチャ。A/Dコンバータの積分時間を秒単位で指定します。

構文 **APER [*aperture*]**

aperture A/D コンバータの積分時間を指定して、前に指定した積分時間あるいは分解能を上書きします。アパーチャの有効範囲は、100ns刻みで0〜1sです(500ns未満の値を指定すると、最小アパーチャの500nsが選択されます)。

電源投入時の***aperture*** = NPLCの電源投入時の値によって決まり、電源ライン周波数が60Hzの場合は166.667msの積分時間が、電源ライン周波数が50Hzまたは400Hzの場合は200msの積分時間が指定されます。
デフォルトの***aperture*** = 500ns

備考

- APERコマンドとNPLCコマンドはどちらも積分時間を設定するため、一方を実行すると、もう一方によって設定済みの積分時間はキャンセルされます。RESコマンド、またはファンクション・コマンドあるいはRANGEコマンドの*%_resolution*パラメータを使用しても、間接的に積分時間を選択できます。分解能を指定する場合、APER(またはNPLC)との関係で次のようになります。

  分解能を指定する**前に**APER(またはNPLC)コマンドを送ると、マルチメータはより大きな分解能(より長い積分時間)を指定するコマンドに従います。

  分解能を指定した**後に**APER(またはNPLC)コマンドを送ると、そのAPER(またはNPLC)コマンドにより指定された積分時間が使用され、前に指定した分解能は無視されます。

- **問合せコマンド** APER?問合せコマンドは、現在指定されている、A/Dコンバータが使用する積分時間(単位秒)を返します。積分時間は、APER、NPLCあるいはRESコマンド、またはファンクション・コマンドあるいはRANGEコマンドの*%_resolution*パラメータによって指定できます。詳細については、本章の始めにある「問合せコマンド」を参照してください。

- **関連コマンド**: FUNC、NPLC、RANGE、RES

例

```
OUTPUT 722; "APER 10E-6" !アパーチャを10μsに設定します。
```

ARANGE

オートレンジ。オートレンジ機能をオンまたはオフにします。

構文 **ARANGE [*control*]**

control *control*パラメータの選択肢は次のとおりです。

control パラメータ	数値 問合せの 等価値	説明
OFF	0	オートレンジ・アルゴリズムをオフにします。
ON	1	オートレンジ・アルゴリズムをオンにします
ONCE	2	マルチメータを1回オートレンジにした後、オートレンジをオフにします。

電源投入時の*control* = ON
デフォルトの*control* = ON

備考

- オートレンジがオンの場合、マルチメータが個々の読み取りの前に入力信号をサンプリングして、適切なレンジを選択します。
- 各測定ファンクションのレンジ一覧は、FUNCまたはRANGEコマンドを参照してください。
- 直接またはサブ・サンプリング測定の場合(DSAC、DSDC、SSACまたはSSDCコマンド)、またはTIMERサンプル・イベントあるいはSWEEPコマンドを使用している場合には、オートレンジは動作しません。
- **問合せコマンド**　ARANGE?問合せコマンドは、現在のオートレンジ・モードを示す応答を返します。詳細については、本章の始めにある「問合せコマンド」を参照してください。
- **関連コマンド**: FUNC, RANGE

例

```
OUTPUT 722;"ARANGE OFF" !オートレンジをオフにします。
```

AUXERR?

補助エラー。ハードウェア・エラーが検出されると、マルチメータが補助エラー・レジスタ内のビットを設定します。AUXERR?コマンドは、設定されたすべてのビットの10進重み和を表す数値を返します。その後、レジスタがクリアされます。

構文 **AUXERR?**

補助エラー条件 補助エラー条件とその重み値は次のとおりです。

重み値	ビット番号	説明
1	0	スレーブ・プロセッサが応答しません。
2	1	DTACK不良
4	2	スレーブ・プロセッサ・セルフテスト不良
8	3	アイソレータ・テスト不良
16	4	A/Dコンバータ・コンバージェンス不良

重み値	ビット番号	説明
32	5	校正値がレンジ外
64	6	GPIBチップ不良
128	7	UART不良
256	8	タイマ不良
512	9	内部過負荷
1024	10	ROMチェックサム不良、下位バイト
2048	11	ROMチェックサム不良、上位バイト
4096	12	不揮発性RAM不良
8192	13	オプションRAM不良
16384	14	Cal RAM書き込みまたは保護不良

備考

- 補助エラー・レジスタは、ハードウェア関連のエラーを示します。1つ以上のビットが設定されていれば、マルチメータは校正または修理が必要です。
- AUXERR?コマンドは、設定されたエラー・ビットがなければ0を返します。
- 補助エラー・レジスタのビットのいずれかが設定されていれば、マルチメータがエラー・レジスタのビット0(ハードウェア・エラー)を設定します。補助エラー・レジスタを読み取っても、エラー・レジスタの0ビットはクリアされません。これをクリアするには、エラー・レジスタを読み取る(ERR?コマンド)必要があります。
- 補助エラー・レジスタのビットをマスクして、これらのビットによるエラー・レジスタのビット0の設定を防ぐことはできません。
- **関連コマンド**: EMASK、ERR?、ERRSTR?、TEST

例

```
10 OUTPUT 722;"AUXERR?"     !補助エラー・レジスタを読み取ります。
20 ENTER 722;A              !重み和を変数Aに入力します。
30 PRINT A                  !重み和を出力します。
40 END
```

例として、AUXERR?コマンドが重み和3072を返したと仮定します。これは、重み値の1024(ROMチェックサム、上位バイト)と2048(ROMチェックサム、下位バイト)にエラーが発生したことを意味します。

AZERO

オートゼロ。オートゼロ機能をオンまたはオフにします。オートゼロ機能は、DC電圧、DC電流および抵抗の測定のみが対象です。

構文 **AZERO [*control*]**

control *control*パラメータの選択肢は次のとおりです。

Control パラメータ	数値 問合せの 等価値	説明
OFF	0	ゼロ測定が1回更新され、その後はファンクション、レンジ、アパーチャ、NPLCまたは分解能の変更後にのみ更新されます。
ON	1	ゼロ測定が各測定後に更新されます。
ONCE	2	ゼロ測定が1回更新され、その後はファンクション、レンジ、アパーチャ、NPLCまたは分解能の変更後にのみ更新されます。

電源投入時の***control*** = ON
デフォルトの***control*** = ON

備考

- オートゼロがONの場合、マルチメータは個々の読み取り後にゼロ測定(入力をオフにした測定)を行い、ゼロ測定値を読み取り値から代数的に減算します。このため、読み取りあたりの時間は約2倍になります。
- *control*パラメータのOFFとONCEは、同じ結果になることに注意してください。オートゼロがOFFまたはONCEであれば、マルチメータはゼロ測定を1回行い、これをその後の読み取り値から代数的に減算します。AZERO OFFまたはAZERO ONCEを実行すると、マルチメータはすべてのイベントのうち、最初のトリガ・アーム・イベントが発生した時点でオートゼロ測定を行います。ただしTARM EXTは例外で、この場合はTARM EXTコマンドが実行されるとゼロ測定が行われます。オートゼロ測定値は、測定ファンクション、レンジまたは積分時間が変更されると常に更新されます(トリガ・アーム・イベントが発生するか、TARM EXTが実行されるとこの更新が行われます)。
- オートゼロがオフの場合は、ディスプレイ・インジケータ**ZERO OFF**が点灯します。
- DC電流測定の場合は、オートゼロはオフにできません。
- オフセット補正をオンにした2端子抵抗測定の場合、ゼロ測定とオフセット測定は同時に行われます。
- 4端子抵抗測定では、オートゼロをオンにしてください。オートゼロをオフにする必要のある場合には、すべての配線を確認にした後でオートゼロをオフにし、リード抵抗が変化しないようにしてください。4端子接続を行う前にオートゼロをオフにするか、(スキャニング時などで)オートゼロをオフにした状態でリード抵抗が変動すると、4端子抵抗測定値は不正確になります。
- **問合せコマンド**　AZERO?問合せコマンドは、現在のオートゼロ・モードを返します。詳細については、本章の始めにある「問合せコマンド」を参照してください。
- **関連コマンド**: DCI、DCV、FUNC、OHM、OHMF

例

```
OUTPUT 722; "AZERO OFF" !DISABLES AUTOZERO
```

BEEP

マルチメータの警報器を制御します。オンの場合、エラーが発生すると1kHzのビープ音を出します。

構文 **BEEP [*control*]**

control *control*パラメータの選択肢は次のとおりです。

control パラメータ	数値 問合せの 等価値	説明
OFF	0	警報器をオフにします。
ON	1	警報器をオンにします。
ONCE	2	1回ビープ音を出し、前のモードに戻ります(OFFまたはONのいずれか)

電源投入時の*control* = 最後にプログラムされた値
デフォルトの*control* = ONCE

備考

- マルチメータは、*control*パラメータを不揮発性メモリに保存します(電源が遮断されても、パラメータは失われません)。
- **問合せコマンド** BEEP? 問合せコマンドは、現在の警報器モードを返します。詳細については、本章の始めにある「問合せコマンド」を参照してください。
- **関連コマンド**: TONE

例

```
OUTPUT 722;"BEEP OFF" !警報器をオフにします。
```

CAL

これは校正コマンドです。詳細については、3458校正マニュアルを参照してください。

CALL

サブプログラムの呼び出し。保存済みのサブプログラムを実行します。

構文 CALL [*name*]

name サブプログラム名。サブプログラム名の最大文字数は10です。名前には、英字、英数字または0～127の範囲の整数が使えます。詳細については、SUBコマンドを参照してください。

電源投入時の*name* = なし

デフォルトの***name*** = 0

備考

- サブプログラムは、SUBコマンドを使用して作成されています。
- マルチメータは､保存済みサブプログラムを実行した後でステータス･レジスタのビット0を設定します。
- 前面パネルから、CALLコマンドにアクセスし、上または下矢印キーを押すことで、すべての保存済みサブプログラムの名前を表示できます。正しいサブプログラムが見つかったら、**Enter**キーを押してそのサブプログラムを実行します。
- **関連コマンド**: COMPRESS、CONT、DELSUB、PAUSE、SCRATCH、SUB、SUBEND

例

```
OUTPUT 722;"CALL DCCUR2"   !"DCCUR2"という名前のサブプログラムを実行します。
```

CALNUM?

校正回数問合せ　マルチメータの校正回数を示す整数を返します。

構文 CALNUM?

備考

- 校正回数は､マルチメータが校正されるごとに1ずつ大きくなります｡自動校正にセキュリティがかかっている場合も､自動校正が実行されるごとに校正回数は1ずつ大きくなります。セキュリティがかかっていない場合、自動校正は校正回数に影響しません。
- 校正回数は、校正保護メモリに保存され、電源が切断されても失われません。
- マルチメータは校正されて出荷されています。マルチメータがお手元に届いたら、校正回数を読み取ってその初期値を確認してください。
- **関連コマンド**: CAL、CALSTR、SCAL

例

```
10 OUTPUT 722;"CALNUM?"        !校正回数を読み取ります。
20 ENTER 722;A                 !応答をコンピュータに入力します。
30 PRINT A                     !応答を出力します。
40 END
```

CALSTR

校正文字列(リモートのみ)　文字列をマルチメータの不揮発性校正RAMに保存します。通常この文字列は、校正時のマルチメータの内部温度(TEMP? コマンド)、校正日、作業者名、次の校正予定日などに使用します。

構文 **CALSTR *string*[,*security_code*]**

string これは、校正RAMに追加される英字/数字メッセージです。*string*パラメータは、一重または二重引用符で囲む必要があります。文字列の最大長は75文字です(文字列を囲む引用符は文字列として数えません)。

security_code 校正RAMにセキュリティがかかっている場合(SECUREコマンド)、校正RAMにメッセージを書き込むためには*security_code*を含める必要があります(文字列を読み取ることは、セキュリティ・モードに関係なくCALSTR?コマンドを使用すれば常に可能です)。校正RAMのセキュリティ保護とセキュリティ解除については、SECUREコマンドを参照してください。

備考

- **問合せコマンド**　CALSTR?問合せコマンドは、マルチメータの校正RAMの文字列を返します。これを、下の2番目の例に示します。
- **関連コマンド**: CAL、CALNUM?、SCAL、SECURE

例 CALSTR

```
OUTPUT 722;"CALSTR 'CALIBRATED 04/02/1987'"
```

CALSTR?

```
10 DIM A$[80]                  !文字列変数の配列を宣言します。
20 OUTPUT 722; "CALSTR?"       !文字列を読み取ります。
30 ENTER 722;A$                !文字列を入力します。
40 PRINT A$                    !文字列を出力します。
50 END
```

COMPRESS

圧縮サブプログラム　メモリに保存済みのサブプログラムからASCIIテキストを削除します。これによりメモリ・スペースを節約できますが、サブプログラムが不揮発性メモリから削除されます(電源が切断されると、サブプログラムは破棄されます)。

構文 **COMPRESS** *name*

name サブプログラム名　サブプログラム名の最大文字数は10です。名前には、英字、英数字または0～127の範囲の整数が使えます。詳細については、SUBコマンドを参照してください。

電源投入時の*name* = なし
デフォルトの*name* = なし、パラメータが必要

備考

- メモリの断片化を防ぐために、他のサブプログラムをダウンロードする前に、個々のサブプログラムを圧縮してください。
- COMPRESSコマンドをサブプログラムの一部として保存することはできません。

- **関連コマンド**: CALL、CONT、DELSUB、PAUSE、SCRATCH、SUB、SUBEND

例

次のプログラム文は、サブプログラムTEST12(ダウンロード済み)を圧縮します。

```
OUTPUT 722;"COMPRESS TEST12"
```

CONT

再開。PAUSEコマンドにより中断されたサブプログラムの実行を再開します。

構文 **CONT**

備考
- GPIB Group Execute Trigger機能も、中断されたサブプログラムの再開に使用できます。
- 中断ステートが継続されるサブプログラムは1つだけです。あるサブプログラムの中断中に、実行されている別のサブプログラムが中断されると、最初のサブプログラムは終了し、2つめのサブプログラムは中断された状態になります。
- **関連コマンド**: PAUSE、SUB、SUBEND

例
```
OUTPUT 722;"CONT”    !サブプログラムの実行を再開します。
```

CSB

クリア・ステータス・バイト　ステータス・レジスタのすべてのビットをクリア(0に設定)します。

構文 **CSB**

備考
- ステータス・レジスタのビットを設定する条件がまだ存在する場合は、CSBコマンドの実行直後に再度そのビットが設定されます。
- ビット6(サービス要求)をクリアすると、マルチメータがGPIB SRQ行を偽に設定します。
- **関連コマンド**: RQS、SPOLL (GPIB コマンド)、STB?

例

```
OUTPUT 722;"CSB"    !ステータス・レジスタをクリアします。
```

DCI、DCV

FUNCコマンドを参照してください。

DEFEAT

マルチメータの入力保護アルゴリズム(下の注意を参照)と、一部の構文およびエラー・チェック・アルゴリズムをオンまたはオフにします。これらのアルゴリズムがオフにされていれば、オンにされている場合よりマルチメータが速く新しい測定構成に移れます。

構文 **DEFEAT [*mode*]**

mode *mode*パラメータの選択肢は次のとおりです。

mode パラメータ	数値 問合せの 等価値	説明
OFF	0	保護、構文およびエラー・アルゴリズムをオンにします。
ON	1	保護、構文およびエラー・アルゴリズムをオフにします。

電源投入時の***mode*** = OFF
デフォルトの***mode*** = OFF

備考

注意 DEFEAT ONは、10V以下のレンジで、入力端子の過負荷電圧が±100Vpeakを超えないことが確実な場合にのみ使用してください(100Vおよび1000Vレンジでは、DEFEATがONかOFFかに関係なく、マルチメータは±1200Vpeakの電圧に耐えられます)。DEFEAT ONでは、マルチメータの入力回路を過負荷電圧から保護する入力スイッチ・シーケンス設定がオフに(無視)されます。入力保護がオフされているときに、10V以下のレンジで過負荷ステートが検出されると、マルチメータが入力保護をオンにして、測定器の保証を考慮するために過負荷に内部的に対応します。

- DEFEAT ONでは、特定の構文チェック・アルゴリズムとエラー・レポート・アルゴリズムがオフされるため、すべてのシステム・プログラムが完成して動作した後にのみ使用してください。
- **問合せコマンド** DEFEAT?問合せコマンドは、現在のDEFEATモードを返します。詳細については、本章の始めにある「問合せコマンド」を参照してください。

例

```
OUTPUT 722;"DEFEAT ON"   !保護、構文およびエラー・アルゴリズムをオフにします。
```

DEFKEY

定義キー。1つ以上のコマンドを前面パネルの特定のユーザ定義ファンクション・キーに割り当てることができます(これらのキーには、f0～f9のラベルが付いています)。1つ以上のコマンドを1つのキーに割り当てた後にそのキーを押すと、割り当てたコマンドがマルチメータのディスプレイに表示されます。**Enter**キーを押すと、リストされた順にコマンドが実行されます。DEFKEY DEFAULTコマンドは、すべてのユーザ定義キーに割り当てられた文字列を消去します。

構文 **DEFKEY** *number,string*

または

DEFKEY DEFAULT

number *number*パラメータは、特定のファンクション・キーを表す0～9の範囲の整数です。

電源投入時の*number* = なし
デフォルトの*number* = 0

string *string*パラメータは、ファンクション・キーに割り当てられたコマンドまたはコマンド・リスト(複数のコマンドをセミコロンでリンクしたもの)です。*string*パラメータは、一重または二重引用符で囲む必要があります。文字列の最大長は40文字です(文字列を囲む引用符は文字列として数えません)。

電源投入時の*string* = なし

デフォルトの*string* = なし(前の**文字列**をクリアします)

DEFAULT

すべてのユーザ定義キーに割り当てられた文字列を消去します。

備考

- 前面パネルから保存した定義は、前面パネルから編集できます。リモートから保存した定義は、編集できません。

- DEFKEY**文字列**に引用符を埋め込むことはできません。このため、引用符で囲んだメッセージを持つDISPコマンドは、*string*パラメータとして使用できません。ただし、DISPコマンドと引用符のないメッセージは使用できます(引用符のないメッセージの制限については、DISPコマンドを参照してください)。

- **問合せコマンド**　DEFKEY?問合せコマンドは、特定のファンクション・キーに現在割り当てられている*string*パラメータを返します(下の例を参照)。DEFKEY?問合せコマンドにより返される文字列は、指定の際に一重引用符または二重引用符のどちらが使用されたかに関係なく、二重引用符で囲まれます。

- **関連コマンド**: LOCK、MENU

例 DEFKEY

```
OUTPUT 722;"DEFKEY 1,'DCI 1;AZERO OFF;NPLC 0'" !コマンドをF1に割り当てます。
```

すべてのDEFKEYのクリア

```
OUTPUT 722;"DEFKEY DEFAULT"  !すべてのDEFKEYをクリアします。
```

DEFKEY?

```
10 OUTPUT 722;"DEFKEY? 1"         !キー1の定義を返します。
20 ENTER 722;A$                   !定義をA$変数に入力します。
30 PRINT A$                       !定義を出力します。
40 END
```

上のプログラムにより返される代表的な応答は、*"DCI 1;AZERO OFF;NPLC 0"*です。DEFKEY 1に何も割り当てられていない場合、上のプログラムは*"EFKEY F1"*を返します。

DELAY

DELAYコマンドを使用すれば、トリガ・イベントと最初のサンプル・イベントの間に挿入される時間間隔を指定できます。

構文 DELAY [*time*]

time 遅延時間を秒単位で指定します。遅延時間の範囲はE-7(100ns)6000秒で、直接サンプリングまたはサブサンプリング(DSAC、DSDC、SSACまたはSSDC)では10ns刻みで、他の全てのファンクションでは100ns刻みです。遅延に対して0を設定すると、遅延が最小値に設定されます。

電源投入時の*time* = 自動(ファンクション、レンジ、分解能およびACBAND設定により決まります)

デフォルトの*time* = 自動(ファンクション、レンジ、分解能およびACBAND設定により決まります)

備考

- デフォルトの遅延は、測定ファンクション(DCV、ACVなど)、レンジ、分解能またはAC帯域幅設定(ACBANDコマンド)を変更するたびに自動的に変化します(ただし、別の値を指定していない場合)。
- **問合せコマンド** DELAY? 問合せは、現在の秒単位の遅延時間を返します。詳細については、本章の始めにある「問合せコマンド」を参照してください。
- **関連コマンド**: NRDGS、SWEEP、TIMER、TRIG

例

```
OUTPUT 722;"DELAY 5"        !5秒の遅延を挿入します。
OUTPUT 722;"DELAY -1"       !自動(デフォルト)遅延に戻ります。
```

DELSUB

サブプログラムの削除　1つのサブプログラムをメモリから削除します。

構文　**DELSUB** *name*

name　**サブプログラム名**　サブプログラム名の最大文字数は10です。名前には、英字、英数字または0～127の範囲の整数が使えます。詳細については、SUBコマンドを参照してください。

電源投入時の*name* = なし
デフォルトの*name* = なし。パラメータが必要

備考
- サブプログラムが削除されると、これを保存するために使用されていたメモリが開放され、新しいサブプログラムの保存に使えるようになります(SUBコマンドを参照)。
- すべてのサブプログラムを一度に削除するには、SCRATCHコマンドを使用します。
- **関連コマンド**: COMPRESS、SCRATCH、SUB

例
```
OUTPUT 722;"DELSUB TEST12"   !サブプログラムTEST12を削除します。
```

DIAGNOST

これはサービス関連コマンドです。詳細については、3458Aサービス・マニュアルを参照してください。

DISP

ディスプレイ　マルチメータのディスプレイをオンまたはオフにし、ディスプレイへのメッセージの送信またはディスプレイのクリアにも使用できます。

構文　**DISP [*control*] [,*message*]**

control　*control*パラメータの選択肢は次のとおりです。

control パラメータ	数値問合せの等価値	説明
OFF	0	メッセージが含まれていれば、表示します(メッセージがない場合は、ダッシュが表示されます)。ERR以外のすべてのインジケータをオフにします。読み取り値は表示されなくなり、前面パネルのキー操作と問合せコマンドを除いて更新されません。
ON	1	ノーマル(電源投入時のモード)ディスプレイ動作
MSG	2	メッセージを表示し、インジケータがオンになります。
CLR	3	ディスプレイをクリアします。

電源投入時の***control*** = ON
デフォルトの***control*** = ON

message *message*パラメータは、表示するメッセージです。メッセージには、スペース、数字、大文字または小文字の英字、および次の文字を含めることができます。

! # $ % & ' ( ) ^ \ / @ ; : [ ] , . + - = * < > ? _

備考

- メッセージにスペース、カンマまたはセミコロンが含まれる場合にのみ、メッセージを引用符で囲む必要があります。一重または二重引用符('または")が使用できます。開始および終了の引用符は同じでなければなりません。
- メッセージの最大文字数は75です(メッセージを囲む引用符は文字数として数えません)
- **問合せコマンド**　DISP?問合せコマンドは、現在指定されている*control*パラメータを返します。詳細については、本章の始めにある「問合せコマンド」を参照してください。
- **関連コマンド**: NDIG

例 次のコマンドでは、マルチメータがメッセージTIME-OUTを表示して、ディスプレイの自動更新を停止します。

```
OUTPUT 722;"DISP OFF,TIME-OUT" !メッセージ = TIME-OUT
```

次のコマンドでは、メッセージにスペースが含まれるため、引用符で囲む必要があります。

```
OUTPUT 722;"DISP MSG,'TIME OUT'" !MESSAGE = TIME OUT
```

DSAC、DSDC

直接サンプリング　マルチメータを直接サンプリング測定(ディジタイズ)用に構成します。DSACファンクションは、入力波形のAC成分だけを測定します。DSDCファンクションは、AC成分とDC成分の合成を測定します。それ以外は、この2つのファンクションは同じです。DSACおよびDSDCファンクションは、トラック/ホールド回路(2nsアパーチャ)と広帯域入力パス(12MHz帯域幅)を使用します。

構文 **DSAC [*max._input*] [,*%-resolution*]**

DSDC [*max._input*] [,*%-resolution*]

max._input 測定レンジを選択します(直接サンプリング測定の場合はオートレンジは使用できません)。レンジを選択するには、*max._input*を入力信号の予想ピーク振幅としてユーザが指定します。こうすると、マルチメータが、正しいレンジを選択します。次の表は、*max._input*パラメータと、マルチメータが選択するレンジを示したものです。

		フル・スケール	
max._put パラメータ	選択される レンジ	SINT フォーマット	DINT フォーマット
0～0.012	10mV	12mV	50mV
>0.012～0.120	100mV	120mV	500mV
>0.120～1.2	1V	1.2V	5.0V
>1.2～12	10V	12V	50V
>12～120	100V	120V	500V
>120～1E3	1000V	1050V	1050V

電源投入時の*max._input*= 該当せず
デフォルトの*max._input* = 10V

%_resolution DSACまたはDSDCコマンドとともに使用するとマルチメータに無視されます。このパラメータは、他のファンクション・コマンド(FUNC、ACT、DCVなど)との一貫性を持たせるために、このコマンド構文で使用できるようになっています。

備考

- 直接サンプリング測定の場合はオートレンジは使用できません。DSACまたはDSDCコマンドの最初のパラメータ(*max._input*パラメータ)としてレンジを指定する必要があります。

- DINTメモリ/出力フォーマットを使用する場合、直接サンプリングのフル・スケール値は、10mV、100mV、1V、10Vおよび100Vの500%(5倍)になることに注意してください。これは、レベル・トリガのパーセントを指定する場合に特に重要です。レベル・トリガ電圧を指定する場合は、レンジのパーセントを使用します。例えば、入力信号のピーク値が20Vで、10Vレンジを使用していると仮定します。15Vでレベル・トリガさせたければ、レベル・トリガのパーセントを150%に指定します(LEVEL 150コマンド)。(周波数が2MHzを超え、振幅がレンジの120%を超える信号を測定すると、マルチメータの増幅器のスルー・レートを超えてしまう可能性があります。周波数が12MHz以下でレンジの120%未満の信号であれば、スルー・レート誤差は生じません)

- マルチメータのトリガ階層(トリガ・アーム・イベント、トリガ・イベントおよびサンプル・イベント)が、直接サンプリングに適用されます。これは、これらのイベントが正しい順序で発生しなければ直接サンプリングが開始されないということです。トリガ階層の詳細については、第4章を参照してください。直接サンプリングでは、TIMERサンプル・イベントとNRDGS n,TIMERコマンド、またはSWEEPコマンドのいずれかを使用できます(SWEEPは、プログラミングがより簡単です)。NRDGSおよびSWEEPコマンドは互換性があり、マルチメータは後に指定されたコマンドを使用します(SWEEPコマンドを使用する場合は、サンプル・イベントが自動的にTIMERに設定されます)。

- 直接サンプリングでは、入力信号のピーク値が指定したレンジの120%未満の場合は、SINTメモリ/出力フォーマットを使用してください。DINTメモリ/出力フォーマットは、入力信号がレンジの120%以上の場合に使用します(SINTおよびDINTは、A/Dコンバータにより内部的に使用されるフォーマットです)。正しいメモリ/出力フォーマットを使用することで、フォーマット変換は不要になります)。

- **関連コマンド**: DSDC、FUNC、LEVEL、LFILTER、SLOPE、NRDGS、PRESET FAST、PRESET DIG、SSAC、SSDC、SSPARM?、SWEEP、TARM、TIMER、TRIG

例 次のプログラムは、DC結合直接サンプリングのデジタイズの例です。SWEEPコマンドは、30μsの間隔と、200個のサンプルを指定しています。レベル・トリガは、10Vレンジの250%に設定されています(10Vの250% = 25V)。サンプルは、読み取り値メモリにDINTフォーマットで送られます。次に、サンプルはコントローラへ送られ、変換され、出力されます。行110を削除すれば、読み取り値メモリを使用しないで、サンプルが直接コントローラに転送されます。ただし、コントローラとGPIBがサンプルを少なくても134kバイト/sのレートで転送可能でなければなりません。これが可能でないと、マルチメータがTRIGGER TOO FASTエラーを生成します。詳細については、第4章の「GPIBによる高速転送」を参照してください。

```
 10 OPTION BASE 1                    !配列の番号付けは1から始まります。
 20 INTEGER Num_samples,I,J,K        !整数変数を作成します。
 30 Num_samples = 200                !200サンプル
 30 ASSIGN @Dvm TO 722               !マルチメータ・アドレスを指定します。
 40 ASSIGN @Buffer TO BUFFER [4*Num_samplesl!バッファI/Oパス名を割り当てます。
 45 !サンプル、(4バイト/サンプル * 200サンプル = 800バイト)
 50 ALLOCATE REAL Samp(1:Num_samples)!サンプル用の実数配列を作成します。
 60 OUTPUT @Dvm;"PRESET FAST"        !DINTフォーマット、TARM SYN、TRIG AUTO
 70 OUTPUT @Dvm;"SWEEP 30E-6,200"    !30μs間隔、200サンプル
 80 OUTPUT @Dvm;"DSDC 10"            !直接サンプリング、10Vレンジ
 90 OUTPUT @Dvm;"LEVEL 250, DC"      !レンジの250%(25V)でレベル・トリガ
100 OUTPUT @Dvm;"TRIG LEVEL"         !レベル・トリガ・イベント
110 OUTPUT @Dvm;"MEM FIFO"           !読み取り値メモリをFIFOモードでオンにします。
120 TRANSFER @Dvm TO @Buffer;WAIT    !サンプルをコントローラに転送します。
130 OUTPUT @Dvm;"ISCALE?"            !DINTフォーマットのスケール・ファクタを問い合わせます。
140 ENTER @Dvm;S                     !スケール・ファクタを入力します。
150 FOR I=1 TO Num_samples
160 ENTER @Buffer USING "#,W,W";J,K    !16ビットの2の補数ワードを1つ
161 !各変数JおよびKに入力します(# = 文ターミネータ不要
161 !W = データを16ビットの2の補数整数として入力)
165 !REQUIRED; W      = ENTER DATA AS 16-BIT 2'S COMPLEMENT INTEGER)
170 Samp(I)=(J*65536.+K+65536.*(K<O))!実数に変換します。
180 R=ABS(Samp(I))                   !絶対値を使用してOVLDがないか確認します。
190 IF R>2147483647 THEN PRINT "OVLD"!過負荷が生じた場合には、メッセージを出力します。
200 Samp(I)=Samp(I)*S                !スケール・ファクタを適用します。
210 Samp(I)=DROUND(Samp(1),8)        !変換された読み取り値を丸めます。
220 PRINT Samp(I)                    !読み取り値を出力します。
230 NEXT I
240 END
```

EMASK

エラー・マスク　特定の条件により、ステータス・レジスタのエラー・ビット(ビット5)が設定されるようにします。

構文 **EMASK [*value*]**

value エラー条件は、その10進重みを*value*パラメータとして指定することでオンにします。複数のエラー条件をオンするには、重みの和を指定します。エラー条件とその重みは次のとおりです。

重み	ビット番号	エラー条件
1	0	ハードウェア・エラー (詳細はAUXERR?を参照)
2	1	校正エラー
4	2	TRIGGER TOO FASTエラー
8	3	構文エラー
16	4	リモートから使用できないコマンド(ADDRESSコマンド)
32	5	未定義パラメータ受信
64	6	パラメータが範囲外
128	7	メモリ・エラー
256	8	破壊的な過負荷検出
512	9	校正が適合しない
1024	10	校正が必要
2048	11	設定の競合(サブサンプリングのためのメモリ構成が不適切)
4096	12	演算エラー (0による割り算、整数オーバフローなど)
8192	13	サブプログラム・エラー (削除されたサブプログラムの呼び出し。PAUSEがないCONT、SUBENDまたはPAUSEはサブプログラム内でのみ使用可能。SCRATCH、DELSUB、CONTはサブプログラム内では使用不可)
16384	14	システム・エラー

電源投入時の*value* = 32767(すべてオン)
デフォルトの*value* = 32767(すべてオン)

備考

- エラーが発生すると、このエラーにより**エラー・レジスタ**の対応するビットが設定されます。これはエラー・レジスタがEMASKコマンドによりオンにされているかどうかには関係ありません。エラー・ビットをオフにすると、**ステータス・レジスタ**のそのエラー・ビットのみが設定されるのを防ぐことができます。したがってサービス要求が生成されます。

- **問合せコマンド**　EMASK? 問合せコマンドは、オンにされたすべてのエラー条件の重み和を返します(下の例を参照)。

- **関連コマンド**: AUXERR?、ERR?、ERRSTR?、RQS、STB?

例

```
OUTPUT 722;"EMASK 4   "          !TRIGGER TOO FASTエラーをオンにします
OUTPUT 722;"EMASK 248 "          !エラー 8、16、32、64および128をオンにします。
OUTPUT 722;"EMASK 0   "          !すべてのエラーをオフにします。
10 OUTPUT 722; "EMASK?"          !EMASK値を返します。
20 ENTER 722;A                   !応答を入力します。
30 PRINT A                       !値を出力します。
40 END
```

END

ENDコマンドは、GPIB EOI(End Or Indentify)機能をオンまたはオフにします。

構文 **END [*control*]**

control *control*パラメータの選択肢は次のとおりです。

control パラメータ	数値 問合せの 等価値	説明
OFF	0	EOI行は決して真に設定されません。
ON	1	読み取り値が複数の場合(SWEEPまたはNRDGS >1)、EOI行が真に設定され、最後の読み取り値の最後のバイトが送られます。読み取り値が1つの場合、EOI行が真に設定され、各読み取り値の最後のバイトが送られます。
ALWAYS	2	個々の読み取り値の最後のバイトが送られると、EOI行が真に設定されます。

電源投入時の*control* = OFF
デフォルトの*control* = ALWAYS

備考

- GPIB への ASCII フォーマットの各読み取り値の出力の後に、通常そのつど *cr,lf*(キャリッジ・リターン、改行)が続きます。*cr lf*は、ほとんどのコントローラに対して伝送の終わりを示します。他のフォーマットの読み取り値の出力の場合は、行シーケンスの終わりに*cr lf*はありません。ASCII出力フォーマットを使用しており、複数の読み取り値がRMEMコマンドを使って読み取り値メモリから呼び出された場合には、マルチメータが読み取り値の間にカンマが入れます。この場合、*cr,lf*は、呼び出されるグループの最後の読み取り値の後に1回だけ現れます。読み取り値が直接バスに出力される場合(読み取り値メモリがオフ)、読み取り値が「暗黙の読み取り」を使用して呼び出される場合、または他の出力フォーマットを使用している場合には、カンマは使用されません。

- コンピュータのEOI行に対する応答については、お使いのコンピュータのマニュアルを確認してください。

- 高速モードに対してEND ALWAYSが指定されている場合は、読み取りが行われている間、EOIモードが自動的にONになります。読み取りが完了すると、EOIモードはALWAYSに戻ります。高速モードの詳細については、第4章の「読み取り速度を上げる」を参照してください。

- **問合せコマンド** END?問合せコマンドは、現在のEOIモードを返します。詳細については、本章の始めにある「問合せコマンド」を参照してください。

- **関連コマンド**: OFORMAT

例

```
OUTPUT 722;"END ALWAYS" !GPIB EOIをオンにします。
```

ERR?

エラーの問合せ　エラーが発生すると、そのエラーによりエラー・レジスタのビットが設定され、ディスプレイのERRインジケータが点灯します。ERR?コマンドは、設定されたすべてのビットを表す数値を返し、レジスタをクリアして、インジケータをオフにします。返された数値は、設定されたすべてのビットの重み和です。

構文 **ERR?**

エラー条件 エラー条件とその重みは次のとおりです。

重み	ビット番号	エラー条件
1	0	ハードウェア・エラー (詳細はAUXERR?を参照)
2	1	校正エラー
4	2	TRIGGER TOO FASTエラー
8	3	構文エラー
16	4	リモートから使用できないコマンド(ADDRESSコマンド)
32	5	未定義パラメータ受信
64	6	パラメータが範囲外
128	7	メモリ・エラー
256	8	破壊的な過負荷検出
512	9	校正が適合しない
1024	10	校正が必要
2048	11	設定の競合(サブサンプリングのためのメモリ構成が不適切)
4096	12	演算エラー (0による割り算、整数オーバフローなど)
8192	13	サブプログラム・エラー (削除されたサブプログラムの呼び出し。PAUSEがないCONT、SUBENDまたはPAUSEはサブプログラム内でのみ使用可能。、SCRATCH、DELSUB、CONTはサブプログラム内では使用不可)
16384	14	システム・エラー

備考

- ERR?コマンドは、設定されたエラー・ビットがなければ0を返します。
- ビット0(重み=1)が設定されている場合、詳細については補助エラーレジスタ(AUXERR?コマンド)を参照してください。
- ERR?コマンドを実行すると、ステータス・レジスタのエラー・ビット(ビット5)がクリアされます。
- **関連コマンド**: AUXERR?、EMASK、ERRSTR?

例

```
10 OUTPUT 722;"ERR?   "          !エラー・レジスタを読み取り、クリアします。
20 ENTER 722;A                   !重み和を変数Aに入力します。
30 PRINT A                       !応答を出力します。
40 END
```

ERRSTR?

エラー文字列の問合せ　ERRSTR?コマンドは、エラー・レジスタまたは補助エラー・レジスタの設定された最下位ビットを読み取って、そのビットをクリアします。ERRSTR?コマンドは、カンマで区切られた2つの応答を返します。1番目の応答はエラー番号(100番台 = エラー・レジスタ、200番台 = 補助エラー・レジスタ)で、2番目の応答はエラーを説明するメッセージ(文字列)です。

構文　**ERRSTR?**

備考

- ERRSTR?により返される文字列の最大長は255文字です。
- ERRSTR?コマンドは、レジスタの設定された最下位ビットだけを読み取ってクリアします。レジスタ内で複数のビットが設定されている場合は、ERRSTR?を繰り返し実行して、設定された各ビットを読み取ってクリアする必要があります。設定されたすべてのビットが読み取られてクリアされると(または、どちらのレジスタにも設定されたビットがない場合には)、ERRSTR?コマンドは*0,"NO ERROR"*を返します。補助エラー・レジスタとエラー・レジスタがクリアされると、ステータス・レジスタのエラービット(ビット5)もクリアされます。
- エラー・レジスタのビット0が設定されている場合、これは補助エラー・レジスタの1つ以上のビットが設定されていることを意味します。この場合、ERRSTR?コマンドは、補助エラー・レジスタの設定された個々のビットを最初に読み取ってクリアします。補助エラーがすべて読み取られると、エラー・レジスタのビット0がクリアされ、ERRSTR?コマンドがエラー・レジスタの残りのエラーの読み取りに使用できるようになります。
- **関連コマンド**: AUXERR?、EMASK、ERR?、QFORMAT

例

```
10 OPTION BASE 1              !配列の番号付けは1から始まります。
20 DIM A$[200]                !文字列変数の配列を宣言します。
30 OUTPUT 722;"ERRSTR?"       !エラー・メッセージを読み取ります。
40 ENTER 722; A,A$            !数値をAに、文字列をA$に入力します。
50 PRINT A,A$                 !応答を出力します。
60 IF A>0 THEN GOTO 30        !エラーごとにループします。
70 END
```

EXTOUT

外部出力。リア・パネルの**Ext Out**コネクタの信号(EXTOUT信号)を生成するイベントを指定します。このコマンドは、EXTOUT信号の極性も指定します。

構文　**EXTOUT [*event*][,*polarity*]**

event *event*パラメータの選択肢は次のとおりです。

event パラメータ	数値問合せの等価値	説明
OFF	0	なし。EXTOUTはオフされます。
ICOMP	1	入力完了(A/Dコンバータによる各読み取り値の積分後、または直接サンプリングあるいはサブサンプリングの場合は、トラック・アンド・ホールドによる入力信号の捕捉後に1µsのパルス)
ONCE	2	EXTOUT ONCEコマンドの実行後1µsのパルスを出力します。その後、イベントはOFFになります。
APER	3	アパーチャ波形(A/Dコンバータによる測定が行われていることを示すレベル)
BCOMP	4	バースト完了(読み取り値のグループの後に1µsのパルス)
SRQ	5	ステータス・イベント発生(GPIB SRQをアサートにするようにオンされたステータス・レジスタ・イベントが発生するたびに1µsのパルス)(2番目の備考を参照)
RCOMP	6	読み取り完了(個々の読み取り後に1µsのパルス)

電源投入時の***event*** = ICOMP
デフォルトの***event*** = ICOMP

polarity EXTOUT信号の極性を指定します。次の選択肢があります。:

polarity パラメータ	数値問合せの等価値	説明
NEG	0	立ち下がりTTL信号を生成します。
POS	1	立ち上がりTTL信号を生成します。

電源投入時の***polarity*** = NEG
デフォルトの***polarity*** = NEG

備考

- APERを除くすべてのイベントが、1µsのパルスをEXTOUTコネクタに生成します。APERが選択されている場合は、A/Dのアパーチャ波形が直接出力されます。EXTOUT信号の立ち上がりが、イベントに対する応答です。上記イベントの詳細については、第4章の「EXTOUT信号」を参照してください。

- ステータス・イベントがステータス・レジスタのSRQビットを設定すると、そのビットはクリア(例えば、CSBコマンド)されるまで設定されたままになります。指定されていれば、SRQ(RQSコマンド)をアサートにするようにオンされたステータス・イベントが発生すると、常にEXTOUT SRQパルスが発生します。EXTOUT SRQパルスは、SRQビットが設定されても、常に発生する訳ではありません。オンされたステータス・イベントが発生すると、常に発生します。

- **問合せコマンド**　EXTOUT?問合せコマンドは、カンマで区切られた2つの応答を返します。1番目の応答は、現在指定されているEXTOUTイベントを示します。2番目の応答は、**極性**を示します。詳細については、本章の始めにある「問合せコマンド」を参照してください。

- 関連コマンド: NRDGS、SRQ、STB?、SWEEP、TBUFF

例 `OUTPUT 722;"EXTOUT APER"` !EXTOUTイベントをアパーチャ波形に設定します。

FIXEDZ

FIXEDZコマンドは、DC電圧測定の固定入力抵抗機能をオンまたはオフにします。オンされている場合、マルチメータがすべてのレンジで入力抵抗を10MΩに保持します。これにより、(レンジの変更により)入力抵抗が変化することでDC電圧測定が影響を受けることを防ぎます。

構文 **FIXEDZ [*control*]**

control *control*パラメータの選択肢は次のとおりです。

control パラメータ	数値 問合せの 等価値	説明	入力抵抗 DCV 1V、1V、10V 1000Vレンジ	DCV 100V、レンジ
オフ	0	FIXEDZ	>10 GΩ	10 MΩ
オン	1	FIXEDZ	10 MΩ	10 MΩ

電源投入時の***control*** = OFF
デフォルトの***control*** = ON

備考
- DC電圧測定から2端子または4端子抵抗測定に変更しても、FIXEDZはオンされたままです。マルチメータの入力抵抗は入力端子と並列の10MΩの抵抗であるため、FIXEDZがオンされた状態で抵抗測定を行うと誤差が生じます。

- DC測定から、AC電圧、AC+DC電圧、すべての種類の電流、周波数／周期測定に変更すると、FIXEDZは一時的にオフされます。例えば、FIXEDZがオンされていて、DC電圧測定からAC電圧測定に変更すると、FIXEDZはオフになります。ただし、DC測定に戻ると、FIXEDZは再度オンされます。

- 問合せコマンド　FIXEDZ? 問合せコマンドは、現在の固定抵抗モードを返します。詳細については、本章の始めにある「問合せコマンド」を参照してください。

- 関連コマンド: DCV、FUNC、OHM、OHMF,

例 `OUTPUT 722; "FIXEDZ ON"` !固定インピーダンスをオンにします。

FREQ

周波数　入力信号の周波数を測定するようにマルチメータに命令します。入力信号をAC電圧、AC+DC電圧、AC電流またはAC+DC電流のどれかにFSOURCEコマンドを使用して指定する必要があります。

構文　**FREQ [*max._input*][,*%_resolution*]**

max._input　固定レンジまたはオートレンジを選択します。レンジは、FSOURCEコマンドで指定された入力信号の種類に対応します。つまり、ACVが指定された入力信号の場合、*max._input*パラメータがAC電圧測定レンジを指定します。固定レンジを選択するには、入力信号の予想ピーク値の絶対値(負の値は使用しない)として*max._input*をユーザが指定します。これで、マルチメータが適切なレンジを選択します。各種入力信号に対して使用可能なレンジを示す表は、FUNCコマンドまたRANGEコマンドを参照してください。

オートレンジ・モードを選択するには、*max._input*に対してAUTOを指定するか、デフォルト・パラメータを指定してください。オートレンジ・モードでは、マルチメータが個々の周波数読み取りの前に入力信号をサンプリングして、適切なレンジを選択します。

電源投入時の*max._input* = 該当せず
デフォルトの*max._input* = AUTO

%_resolution　*%_resolution*パラメータは、下に示すように分解能の桁数とゲート時間を指定します(*%_resolution*は、読み取り速度にも影響します。詳細については、付録Aの仕様を参照してください)。

%_resolution パラメータ	選択される ゲート時間	分解能桁数
0.00001	1s	7
0.0001	100ms	7
0.001	10ms	6
0.01	1ms	5
0.1	100µs	4

電源投入時の*%_resolution* = 該当せず
デフォルトの*%_resolution* = 0.00001

備考

- 読み取り速度は、入力信号の 1 周期、ゲート時間、またはデフォルトの読み取りタイム・アウトの1.2秒のうち一番長いものです。

- 周波数(および周期)測定は、 特定の電圧を正または負のスロープで入力信号が交差する時点を検出するレベル検出回路を使用して行われます(周波数または周期測定時には、LEVELトリガ・イベント、LEVELサンプル・イベントまたはLINEトリガ・イベントを使用できないのはこのためです)。電源投入時のレベル・トリガ値またはデフォルトのレベル・トリガ値では、正のスロープのゼロ・ボルトが選択されます。レベル・トリガ電圧と結合は、LEVELコマンドを使用して設定することはできません。SLOPEコマンドを使用すれば、正または負のスロープのいずれかを指定できます。

- 最も左の桁は、ほとんどの測定ファンクションで1/2桁ですが、周波数測定では0から9までのフルの桁に表示されます。

- オートレンジをオンにして行う読み取りは、より長い時間がかかります。これは、周波数読み取りの間に(適切なレンジを決めるため)入力信号がサンプリングされるためです。

- 周波数(および周期)測定では、過負荷表示は電圧または電流の振幅が指定された測定レンジに対して大きすぎることを意味します。印加された周波数(または周期)が大きすぎて測定できないということではありません。

- **関連コマンド**: ACBAND、FSOURCE、FUNC、LFILTER、PER、RES

例

```
10 OUTPUT 722;"FSOURCE ACI"        !AC電流を入力信号源として選択します。
20 OUTPUT 722;"FREQ .01,.001"      !周波数測定、10mAレンジを選択します。
25                                 !10msゲート時間、5桁の分解能を選択します。
30 END
```

FSOURCE

周波数信号源　周波数または周期測定の入力信号として使用する信号の種類を指定します。

構文 **FSOURCE [*source*]**

source *source*パラメータの選択肢は次のとおりです。

source パラメータ	数値問合せの等価値	説明(測定機能)
ACV	2	AC電圧(周波数1Hz～10MHz、周期100ns～1s)
ACDCV	3	AC+DC電圧(周波数1Hz～10MHz、周期100ns～1s)
ACI	7	AC電流(周波数1Hz～100kHz、周期10μs～1s)
ACDCI	8	AC+DC電流(周波数1Hz～100kHz、周期10μs～1s)

電源投入時の***source*** = ACV
デフォルトの***source*** = ACV

備考

- **問合せコマンド**　FSOURCE? 問合せコマンドは、現在の周波数信号源を返します。詳細については、本章の始めにある「問合せコマンド」を参照してください。

- **関連コマンド**: FREQ、FUNC、PER

例

```
10 OUTPUT 722;"FSOURCE ACDCI"      !ACDCIを入力信号源として選択します。
20 OUTPUT 722;"FREQ .l,.Ol"        !周波数、100mAレンジ、1msの
25                                 !ゲート時間、4桁の分解能を選択します。
30 END
```

FUNC

ファンクション　測定の種類(AC電圧、DC電流など)を選択します。測定レンジと分解能を指定することもできます(FUNCヘッダはオプションで、省略できます)。

構文 **FUNC [*function*][,*max._input*][,*%_resolution]***

または

[FUNC] *function*[,*max._input*][,*%_resolution*]

function *function*パラメータは、測定の種類を指定します。パラメータの選択肢は次のとおりです。

function パラメータ	数値問合せの等価値	説明
DCV	1	DC電圧測定を選択します。
ACV	2	AC電圧測定を選択します(このモードは、SETACVコマンドにより設定します)。
ACDCV	3	AC+DC電圧測定を選択します(このモードは、SETACVコマンドにより設定します)。
OHM	4	2端子抵抗測定を選択します。
OHMF	5	4端子抵抗測定を選択します。
DCI	6	DC電流測定を選択します。
ACI	7	AC電流測定を選択します。
ACDCI	8	AC+DC電流測定を選択します。
FREQ ∗	9	周波数測定を選択します。
PER ∗	10	周期測定を選択します。
DSAC ∗	11	直接サンプリング、AC結合
DSDC ∗	12	直接サンプリング、DC結合
SSAC ∗	13	サブサンプリング、AC結合
SSDC ∗	14	サブサンプリング、DC結合

∗これらのファンクションは別に説明が必要ですが、本章の中で個別に記載してあります。詳細については、対応するDSAC、DSDC、FREQ、PER、SSAC、SSDCコマンドを参照してください。

電源投入時の*function* = DCV
デフォルトの*function* = DCV

max._input 固定レンジ・モードまたはオートレンジ・モードを選択します。固定レンジを選択するには、*max._input*を入力信号の最大予想振幅(または、抵抗測定の最大抵抗)の絶対値(負の値は使用しない)としてユーザが指定します。こうすると、マルチメータが正しいレンジを選択します。

オートレンジを選択するには、*max._input*に対してAUTOを指定するか、デフォルト・パラメータを指定します。オートレンジ・モードでは、マルチメータが各読み取りの前に入力信号をサンプリングし、適切なレンジを選択します。

- 次の表は、*max._input*パラメータと各測定ファンクションに対して選択されるレンジを示します。

DCV:

max._input パラメータ	選択されるレンジ	フル・スケール
–1またはAUTO	オートレンジ	
0～0.12	100mV	120mV
>0.12～1.2	1V	1.2V
>1.2～12	10V	12V
>12～120	100V	120V
>120～1E3	1000V	1050V

ACVまたはACDCV:

max._input パラメータ	選択されるレンジ	フル・スケール
–1またはAUTO	オートレンジ	
0～0.012	10mV	12mV
>0.012～0.12	100mV	120mV
>0.12～1.2	1V	1.2V
>1.2～12	10V	12V
>12～120	100V	120V
>120～1E3	1000V	1050V

OHMまたはOHMF:

max._input パラメータ	選択されるレンジ	フル・スケール
–1またはAUTO	オートレンジ	
0～12	10Ω	12Ω
>12～120	100Ω	120kΩ
>120～1.2E3	1kΩ	1.2kΩ
>1.2E3～1.2E4	10kΩ	12kΩ
>1.2E4 1.2E5	100kΩ	120kΩ
>1.2E5～1.2E6	1MΩ	1.20MΩ
>1.2E6～1.2E7	10MΩ	12MΩ
>1.2E7 1.2E8	100MΩ	120MΩ
>1.2E8 1.2E9	1GΩ	1.2GΩ

DCI:

max._input パラメータ	選択されるレンジ	フル・スケール
–1またはAUTO	オートレンジ	
0～0.12E-6	0.1μA	0.12μA
>0.12E-6～1.2E-6	1μA	1.2μA
>1.2E-6～12E-6	10μA	12μA
>12E-6～120E-6	100μA	120μA
>120E-6～1.2E-3	1mA	1.2mA
>1.2E-3～12E-3	10mA	12mA
>12E-3～120E-3	100mA	120mA
>120E-3～1.2	1A	1.05A

ACIまたはACDCI:

max._input パラメータ	選択されるレンジ	フル・スケール
–1またはAUTO	オートレンジ	
0～0.120E-6	100μA	120μA
>120E-6～1.2E-3	1mA	1.2mA
>1.2E-3～12E-3	10mA	12mA
>12E-3～120E-3	100mA	120mA
>120E-3～1.2	1A	1.05A

DSACまたはDSDC:

		フル・スケール	
max._input パラメータ	選択されるレンジ	SINT フォーマット	DINT フォーマット
0～0.012	10mV	12mV	50mV
>0.012～0.120	100mV	120mV	500mV
>0.120～1.2	1V	1.2V	5.0V
>1.2～12	10V	12V	50V
>12～120	100V	120V	500V
>120～1E3	1000V	1050V	1050V

SSACまたはSSDC:

max._input パラメータ	選択されるレンジ	フル・スケール
0～0.012	10mV	12mV
>0.012～0.120	100mV	120mV
>0.120～1.2	1V	1.2V
>1.2～12	10V	12V
>12～120	100V	120V
>120～1E3	1000V	1050V

電源投入時の*max._input* = AUTO
デフォルトの*max._input* = **AUTO**

%_resolution ほとんどの測定ファンクションでは、*%_resolution*を*max._input*パラメータのパーセントとしてユーザが指定します(*%_resolution*の周波数／周期測定に対する影響を示す表は、FREQおよびPERコマンドを参照してください。functionパラメータがDSAC、DSDC、SSACまたはSSDCの場合、*%_resolution*は無視されます)。

FREQ、PER、DSAC、DSDC、SSACおよびSSDCを除くすべてのファンクションでは、マルチメータが*%_resolution*に*max._input*を掛けて測定の分解能を求めます。例えば、DC電圧を測定する場合に、最大予想入力が10Vで、1mVの分解能が必要と仮定します。*%_resolution*を求めるには、次の式を使用してください。

%_resolution = (実際の分解能/最大入力)×100

この例では、式は次のようになります。

%_resolution = (0.001/10)×100 = 0.0001×100 = 0.01

注記 オートレンジを使用する場合、マルチメータは、*%_resolution*パラメータに選択されたレンジのフル・スケール読み取り値を掛けます。その結果が、最小分解能です。常に少なくとも最小分解能になり、多くの場合これに余分の分解能桁数が加わります。

電源投入時の*%_resolution* = なし 電源投入時、分解能はNPLCコマンドによって決まり、この場合8 1/2桁になります(NDIGの電源投入時の値はディスプレイの1桁をマスクするため、マルチメータは7 1/2桁しか表示しません。NDIG 8コマンドを使用すれば、8 1/2桁をすべて表示できます。詳細は、NDIGコマンドを参照してください)。

デフォルトの*% resolution*:

周波数または周期測定では、デフォルトの*%_resolution*は0.00001で、これにより1sのゲート時間と7桁の分解能が選択されます。

サンプリングしたACVまたはACDCVでは、デフォルトの*%_resolution*は、SETACV SYNCでは0.01%、SETACV RNDMでは0.4%です。

他のすべての測定ファンクションでは、デフォルトの分解能は現在の積分時間で決まります。

備考

- **問合せコマンド** FUNC? 問合せコマンドは、カンマで区切られた2つの応答を返します。1番目の応答は現在のファンクションです。2番目の応答は、現在の測定レンジです(これは実際のレンジで、必ずしも*max._input*に対して指定された値ではありません)。FUNC?問合せコマンドは、オートレンジ・モードを示しません。オートレンジ・モードを確認するにはARANGE?を使用してください。詳細については、本章の始めにある「問合せコマンド」を参照してください。

- **関連コマンド**: ACDCI、ACDCV、ACI、ACV、APER、DCI、DCV、DSAC、DSDC、FREQ、OHM、OHMF、PER、RATIO、NPLC、RES、SETACV、SSAC、SSDC

例　次のプログラムでは、行10により、行20の*%_resolution*が分解能を設定することが可能になります。行20が指定する分解能は、6V×0.0000167 = 100µVです。

```
10 OUTPUT 722;"NPLC 0"              !PLCを最小に設定します。
20 OUTPUT 722;"FUNC DCV,6,.00167"  !DC電圧、6V最大
30 END                              !100µV分解能を選択します。
```

次のプログラムでは、行10がPLC数を1000に設定します。これは最大分解能(7.5桁)に相当し、行20の*%_resolution*が測定に影響することを防ぎます。行20が要求する分解能は10µΩです。ただし、行10のために、実際の分解能は100µΩになります。

```
10 OUTPUT 722;"NPLC 1000"           !PLCを最大に設定します。
20 OUTPUT 722;"FUNC OHM,1E3,.001"  !2端子抵抗、
30 END                              !1kΩ最大、10mΩ分解能を選択します。
```

ID?

IDの問合せ　マルチメータはID?コマンドに対して、文字列"HP 3458A"を返すことで応答します。この機能により、GPIBコントローラがマルチメータをそのアドレスで特定することが可能になります。

構文　**ID?**

備考　• 関連コマンド: ADDRESS、QFORMAT

例

```
10 OUTPUT 722;"ID?”         !応答を返します
20 ENTER 722;A$             !応答をコンピュータのA$変数に入力します。$ VARIABLE
30 PRINT A$                 !応答を出力します。
40 END
```

INBUF

入力バッファ　マルチメータの入力バッファをオンまたはオフにします。オンの場合、入力バッファによりGPIBバスにより受信したコマンドを一時的に保存されます。これにより、コマンドが受信されると直ちにバスが開放され、コントローラは他のタスクを実行し、一方マルチメータは保存されたコマンドを実行することができます。

構文　**INBUF [*control*]**

control *control*パラメータの選択肢は次のとおりです。

control パラメータ	数値問合せの等価値	説明
OFF	0	入力バッファをオフにします。コマンドは、マルチメータがビジーでない場合にのみ受け付けられます。
ON	1	入力バッファをオンにします。コマンドが保存され、バスが直ちに開放されます。

電源投入時の*control* = OFF
デフォルトの*control* = ON

備考

- 入力バッファをOFFにすると、速度性能がやや落ちますが、バス・アクティビティと同期させる場合は便利です。入力バッファがOFFの場合、マルチメータは一度に1つのコマンドだけを受け付け、そのコマンドの実行が終了するまでバスを開放しません。このため、他のバス・デバイスに送られた後続のコマンドは、マルチメータがそのコマンドの実行を終了するまで実行されせん。

- 入力バッファをONにすると、マルチメータは受信メッセージをバッファリング(保存)し、メッセージ転送が完了すると直ちにGPIBバスを開放します。このため、マルチメータがそのコマンドを実行している間も、コントローラが他のバス・デバイスと通信することができます。ただし、マルチメータがその命令を終了する前に他のバス・デバイスが命令を実行した場合には、他のバス・デバイスとの同期は失われます。この場合は、ステータス・レジスタのレディ・ビットを(シリアル・ポーリングにより)モニタして、マルチメータの命令完了を判別することができます。

- コマンド群が255文字より長いと入力バッファがいっぱいになり、マルチメータは、最初に受信したコマンドを実行するまでバス･アクティビティを停止します。残りのメッセージは、バッファに余裕ができると入力されます。

- **問合せコマンド**　INBUF?問合せコマンドは現在の入力バッファ・モードを 返します。詳細については、本章の始めにある「問合せコマンド」を参照してください。

例 次のプログラムは、自動校正ルーチンのすべてを実行する前に入力バッファをオンにします。これにより、完了に11分以上かかる自動校正の間、バスがホールドされることがありません。

```
10 OUTPUT 722;"INBUF ON"       !入力バッファをオンにします。
20 OUTPUT 722;"ACAL ALL"       !自動校正(11分以上かかります)
30 END
```

ISCALE?

整数スケールの問合せ　SINTまたはDINTフォーマットの出力を読み取るためのスケール・ファクタを返します。

構文 ISCALE?

備考

- ASCII、SREALおよびDREAL出力フォーマットの場合、スケール･ファクタは常に1です。
- SINTまたはDINTフォーマットの読み取り値出力(OFORMATコマンドを参照)は、整数で表せるように最初にマルチメータにより圧縮されます。読み取り値にISCALE?により返される値を掛けることで、読み取り値の実際の値が復元されます。スケール・ファクタは、ISCALE?実行時のマルチメータの構成で決まります。この構成には、測定ファンクション、レンジ、積分時間などがあります。したがって、スケール・ファクタを取得するときと、読み取りが行われたときのマルチメータの構成が同じでなければなりません。マルチメータを構成した後、スケール・ファクタを取得することができますが、これは読み取りがトリガされる前、または読み取り直後です。
- リアルタイムまたは後処理演算機能がオンの場合(STATまたはPFAILを除く)、あるいはオートレンジがオンされている場合には、周波数／周期測定にSINTまたはDINT出力フォーマット、あるいはメモリ・フォーマットを使用しないでください。
- 関連コマンド: OFORMAT、SSAC、SSDC

例 SINTの例

次のプログラムは、SINTフォーマットで10個の読み取り値を出力し、スケール・ファクタを取得して、各読み取り値にスケール・ファクタを掛けます。

```
10 OPTION BASE 1                 !配列の番号付けは1から始まります。
20 INTEGER Int_rdgs (1:10) BUFFER !整数バッファ配列を作成します。
30 REAL Rdgs(1:10)               !実数配列を作成します。
40 Num_readings=10               !読み取り値の個数 = 10
50 ASSIGN @Dvm TO 722            !マルチメータ・アドレスを割り当てます。
60 ASSIGN @Int_rdgs TO BUFFER Int_rdgs(*)  !バッファ I/Oパス名を割り当てます。
70 OUTPUT @Dvm;"PRESET NORM;OFORMAT SINT;NPLC 0;NRDGS ";Num_readings
75 !TARM AUTO、TRIG SYN、SINT出力フォーマット、最小積分時間
80 TRANSFER @Dvm TO @Int_rdgs;WAIT!SYNイベント、整数配列に読み取り値を
81 !入力します。コンピュータの整数フォーマットはSINTと同じであるため、
85 !ここではデータ変換は不要です(整数配列が必要)。
90 OUTPUT @Dvm;"ISCALE?"         !SINTフォーマットのスケール・ファクタの問合せ
100 ENTER @Dvm;S                 !スケール・ファクタを入力します。
110 FOR I=1 TO Num_readings
120  Rdgs(I)=Int_rdgs(I)         !各読み取り値を実数に変換します。
125  !フォーマット(次の行への整数オーバフローを防ぐために必要)
130  R=ABS(Rdgs(I))              !絶対値を使用してOVLDがないか確認します。
140  IF R>=32767 THEN PRINT "OVLD"!OVLDの場合、過負荷メッセージを出力します。
150  Rdgs(I)=Rdgs(I)*S           !読み取り値にスケール・ファクタを掛けます。
160  Rdgs(I)=DROUND(Rdgs(I),4)   !4桁に丸めます。
170 NEXT I
180 END
```

DINTの例

次のプログラムは、50個の読み取り値を取得し、これをDINTフォーマットを使用してコンピュータに転送すること以外は、前のプログラムと似ています。

```
10 OPTION BASE 1              !配列の番号付けは1から始まります。
20 INTEGER Num_readings,I,J,K  !変数を宣言します。
```

```
30 Num_readings=50              !読み取り値の個数 = 50
40 ALLOCATE REAL Rdgs(l:Num_readings) !読み取り値用の配列を作成します。
50 ASSIGN @Dvm TO 722           !マルチメータ・アドレスを割り当てます。
60 ASSIGN @Buffer TO BUFFER[4*Num_readingsl!バッファ I/Oパス名を割り当てます。
70 OUTPUT @Dvm;"PRESET NORM;RANGE 10;OFORMAT DINT;NRDGS ";Num_readings
75 !ARM AUTO、TRIG SYN、DCV10Vレンジ、DINT出力フォーマット、NRDGS 50、AUTO
80 TRANSFER @Dvm TO @Buffer;WAIT!SYNイベント、読み取り値を転送します。
90 OUTPUT @Dvm; "ISCALE?"       !DINTのスケールを問い合わせます。
100 ENTER @Dvm;S                !スケール・ファクタを入力します。
110 FOR I=1 TO Num_readings
120 ENTER @Buffer USING "#,W,W";J,K  !1つの16ビットの2の補数ワード を
121 !個々の変数JおよびKに入力します(# = 文ターミネータは
125 !不要です。W          = データを、16ビットの2の補数として入力します)。
130 Rdgs(I)=(J*65536.+K+65536.*(K<O))  !実数に変換します。
140 R=ABS(Rdgs(I))              !絶対値を使用してOVLDがないか確認します。
150 IF R>2147483647 THEN PRINT "OVLD"  !過負荷が発生した場合、メッセージを出力します。
160 Rdgs(I)=Rdgs(I)*S           !スケール・ファクタを適用します。
170 Rdgs(I)=DROUND(Rdgs(I),8)  !変換された読み取り値を丸めます。
180 PRINT Rdgs(I)               !読み取り値を出力します。
190 NEXT I
200 END
```

LEVEL

LEVELコマンドは、レベル・トリガ電圧(現在のレンジのパーセントとして)とレベル・トリガの結合(ACまたはDC)を指定します。レベル・トリガ・イベントは、SLOPEコマンドで指定された立ち上がりまたは立ち下がりスロープ上の指定電圧に入力信号が達すると発生します。

構文 **LEVEL [*percentage*][,*coupling*]**

percentage

レベル・トリガのための、現在のレンジのパーセントを指定します。このパラメータの有効範囲は、直接サンプリングまたはサブサンプリングの場合は5%ステップで–500%〜+500%、DC電圧の場合は1%ステップで–120%〜120%です(詳細は第5章を参照してください)。

電源投入時の***percentage*** = 0%(0V)
デフォルトの***percentage*** = 0%(0V)

直接サンプリングのフルスケール値は、10mV 100mV、1V、10Vおよび100Vレンジの500%(5倍)です。レベル・トリガのパーセントを指定する際には、レンジのパーセントを使用してください。例えば、入力信号のピーク値が20Vpで、10Vレンジを使用していると仮定します。15Vでレベル・トリガさせる場合は、レベル・トリガのパーセントとして150%(LEVEL 150コマンド)を指定します。

coupling

couplingパラメータは、信号とレベル検出回路の結合だけを選択します。これは、測定対象の信号の結合には影響しません。

coupling パラメータ	数値問合せの等価値	説明
DC	1	レベル検出回路へのDC結合入力を選択します。
AC	2	レベル検出回路へのAC結合入力を選択します。

電源投入時の***coupling*** = AC
デフォルトの***coupling*** = AC

備考

- レベル・トリガは、DC電圧、直接サンプリングおよびサブサンプリングに対して使用できます(LEVELコマンドは、周波数／周期測定のゼロ交差しきい値と入力信号結合にも影響します)。DC電圧および直接サンプリングの場合、レベル・トリガは、トリガ・イベント(TRIG LEVELコマンド)またはサンプル・イベント(NRDGS n, LEVELコマンド)として使用できます。サブサンプリングの場合は、レベル・トリガは同期信号源イベントのみに対して使用できます(SSRC LEVEL コマンド)。
- ヒステリシスのため、実際のレベル・トリガ・ポイントは、指定した**パーセント**測定レンジの±4%です。
- DC電圧測定に対してレベル・トリガ(AZERO OFF コマンド)を使用する場合には、オートゼロはオフする必要があります(オートゼロは、直接サンプリングまたはサブサンプリングには適用されません)。
- **問合せコマンド**　LEVEL?問合せコマンドは、カンマで区切られた2つの応答を返します。1番目の応答は、現在指定されている**パーセント**です。2番目の応答は、現在の結合モードです。詳細については、本章の始めにある「問合せコマンド」を参照してください。
- **関連コマンド**: DCV、DSAC、DSDC、LFILTER、NRDGS、SETACV、SYNC、SLOPE、SSAC、SSDC、SSRC、TRIG

例

```
10 OUTPUT 722;"TARM HOLD"         !トリガを中断します。
20 OUTPUT 722;"PRESET DIG"        !高速DCV測定、10Vレンジ
30 OUTPUT 722;"TRIG LEVEL"        !レベル・トリガ・イベントを選択します。
40 OUTPUT 722;"SLOPE POS"         !信号の正のスロープでトリガします。
50 OUTPUT 722;"LEVEL 50,AC"       !10Vレンジの50%(5V)、AC結合でトリガします。
60 END
```

LFILTER

レベル・フィルタ　レベル・フィルタ機能をオンまたはオフにします。オンの場合、レベル・フィルタ機能により単極ローパス・フィルタ回路がレベル検出回路の入力に接続されます。ローパス・フィルタの3-dBポイントは75kHzで、高周波成分が誤ったトリガの原因になることを防ぎます。

構文 **LFILTER [*control*]**

control

*control*パラメータの選択肢は次のとおりです。

control パラメータ	数値問合せの等価値	説明
OFF	0	レベル・フィルタをオフにします。フィルタリングは行われません。
ON	1	レベル・フィルタをオンにします。

電源投入時の***control*** = OFF
デフォルトの***control*** = ON

備考
- レベル・フィルタリングは、DC電圧、直接サンプリング、サブサンプリングに対してレベル・トリガを行う場合に使用できます。レベル・フィルタは、周波数／周期測定、または同期法(SETACV SYNCコマンド)を使用してACまたはAC+DC電圧の測定を行う場合に、ノイズに対する感度を下げるためにも使用できます。
- **問合せコマンド**　LFILTER?問合せコマンドは、現在のレベル・フィルタのモードを返します。詳細については、本章の始めにある「問合せコマンド」を参照してください。
- **関連コマンド**: DCV、DSAC、DSDC、FREQ、LEVEL、NRDGS、PER、SETACV、SYNC、SLOPE、SSAC、SSDC、SSRC、TRIG

例
```
OUTPUT 722;"LFILTER ON" !レベル・フィルタをオンにします。
```

LFREQ

LFREQコマンドを使用すれば、A/Dコンバータの基準周波数を指定したり、ライン周波数を測定してその値を基準周波数に設定することができます。

構文 **LFREQ [*frequency*]**

または

LFREQ LINE

frequency

基準周波数を設定できます。**周波数**パラメータの有効範囲は、45～65Hzまたは360～440Hzです。ユーザが360～440Hzの範囲で周波数を指定すると、マルチメータがこの値を8で割ります。例えば、LFREQ 400を指定すると、マルチメータは基準周波数を400/8 = 50Hzに設定します。

電源投入時のreference frequency = 50または60Hzに丸められた値(下の最初の備考を参照)

デフォルトのreference frequency = 実際に測定されたライン周波数(または、ライン周波数が400Hzの場合は測定値/8)

LINE

ライン周波数の実際の値を測定して、基準周波数をこの値に設定します(または、測定値が360Hz～440Hzの場合は、測定値/8)

備考
- 電源を投入すると、マルチメータがライン周波数を測定して、これを50Hzまたは60Hzに丸め、A/Dコンバータの基準周波数をこの丸めた値に設定します(400Hz電源ライン周波数の場合、マルチメータは400Hzの分周波である50Hzを基準周波数として使用します)。

- 基準周波数の周期のステップ・サイズは100nsです。例えば、60Hzの基準周波数の周期は、1/60Hz = 0.0166666..です。ステップ・サイズは100nsであるため、マルチメータ は0.0166667sの値を使用します。ステップ・サイズの最も分かりやすい例は、LFREQ?問合せコマンドを使用する場合です。例えば、60Hzを基準周波数として指定した場合、LFREQ?は59.99988(1/0.0166667)を返します。

- マルチメータは基準周波数の周期に、指定した電源ライン周波数 (NPLC コマンド)を掛けて、実際の積分時間を求めます。マルチメータのDC測定と抵抗測定のノーマル・モード・ノイズ除去(NMR)仕様が、A/Dコンバータの基準周波数の確度に関係します。

- **問合せコマンド**　LFREQ?問合せコマンドは、マルチメータのA/Dコンバータにより使用されている電源ライン周波数基準の現在の値を返します。ステップ・サイズが100nsであるため、指定された値の周期がl/l00nsで均等に割り切れない場合は、LFREQ?により返される値は指定した値とわずかに異なります。詳細については、本章の始めにある「問合せコマンド」を参照してください。

- **関連コマンド**: LINE?、NPLC

例
```
OUTPUT 722; "LFREQ LINE"!電源ライン周波数を測定し、基準周波数
                         !をその測定値(または、電源ライン周波数が400Hzの場合
                         !は値/8)に設定します。
```

LINE?

電源ライン周波数の問合せ　AC電源ラインの周波数を測定して返します。

構文　**LINE?**

備考
- 電源ライン周波数を測定して、A/Dコンバータの基準周波数をその測定値に自動的に設定する方法は、前のページのLFREQコマンドを参照してください。

- **関連コマンド**: LFREQ

```
10 OUTPUT 722; "LINE?"        !電源ライン周波数を測定します。
20 ENTER 722;A                !応答をコンピュータのA変数に入力します。
30 PRINT A                    !応答を出力します。
40 END
```

LOCK

ロックアウト。マルチメータのキーボードをオンまたはオフにします。

構文 **LOCK [*control*]**

control

*control*パラメータの選択肢は次のとおりです。

control パラメータ	数値問合せの等価値	説明
OFF	0	キーボードをオンにします(通常の動作)。
ON	1	キーボードをオフにします(キーを押しても機能しません)。

電源投入時の***control*** = OFF
デフォルトの***control*** = ON

備考
- LOCKコマンドには、前面パネルのアルファベット・コマンド・ディレクトリからアクセスできます。ただし、前面パネルからLOCKコマンドを実行しても機能しません。
- キーボードをオフにした後、オンするにはコントローラを使用するか、電源を入れなおすしかありません。LOCKコマンドは、マルチメータのローカル・キーをオフにします。
- 問合せコマンド　LOCK?問合せコマンドは、現在のLOCKモードを返します。詳細については、本章の始めにある「問合せコマンド」を参照してください。
- 関連コマンド: LOCAL LOCKOUT (GPIB コマンド)

例
```
OUTPUT 722;"LOCK ON" !キーボードをオフにします。
```

MATH

MATHコマンドは、リアルタイム演算をオンまたはオフにします。

構文 **MATH [***operation_a***][,***operation_b***]**

operation

*operation*パラメータの選択肢は次のとおりです。

operation パラメータ	等価数値	説明
OFF	0	オンされているリアルタイム演算をすべてオフにします。
CONT	1	前の演算をオンにします。2つの演算を再開するには、MATH CONT,CONTを送ります。
CTHRM	3	結果 = 5kΩサーミスタ(40653B)の温度(°C)。ファンクションは、OHMまたはOHMF(10kΩレンジ以上)でなければなりません。
DB	4	結果 = $20 \times Log_{10}$(読み取り値/REFレジスタ)。REFレジスタが1に初期化され、dBVとなります。
DBM	5	結果 = $10 \times log_{10}$(読み取り値^2/RESレジスタ/1mW)。ファンクションは、ACV、DCVまたはACDCVでなければなりません。
FILTER	6	結果 = 指数重みデジタル・ローパス・フィルタの出力。応答は、DEGREEレジスタにより設定されます。
FTHRM	8	結果 = 5kΩサーミスタ(40653B)の温度(F)。ファンクションは、OHMまたはOHMF(10kΩレンジ以上)でなければなりません。
NULL	9	結果 = 読み取り値 – OFFSETレジスタ　OFFSETレジスタは、最初の読み取り値に設定されます。その後、これは変更できます。
PERC	10	結果 = ((読み取り値 – PERCレジスタ)/PERCレジスタ×100
PFAIL	11	読み取り値対MAXおよびMINレジスタ
RMS	12	結果 = 読み取り値を2乗し、FILTER演算を適用して、平方根を求めます。
SCALE	13	結果 = (読み取り値 – OFFSETレジスタ)/SCALEレジスタ
STAT	14	現在の読み取り値グループに対して統計計算を実行して、結果を次のレジスタに保存します。 SDEV = 標準偏差 MEAN = 読み取り値の平均 NSAMP = 読み取り値の個数 UPPER = 最大読み取り値 LOWER = 最小読み取り値
CTHRM2K	16	結果 = 2kΩサーミスタ(40653A)の温度(°C)。ファンクションは、OHMまたはOHMFでなければなりません。
CTHRM10K	17	結果 = 10kΩサーミスタ(40653C)の温度(°C)。ファンクションは、OHMまたはOHMFでなければなりません。
FTHRM2K	18	結果 = 2kΩサーミスタ(40653A)の温度(F)。ファンクションは、OHMまたはOHMFでなければなりません。
FTHRM10K	19	10kΩサーミスタ(40653C)の結果=温度(F)。ファンクションは、OHMまたはOHMFでなければなりません。

operation パラメータ	等価数値	説明
CRTD85	20	結果 = αが0.00385の100Ω RTD(40654Aまたは406548)の温度(°C)。ファンクションは、OHMまたはOHMFでなければなりません。
CRTD92	21	結果 = αが0.003916の100Ω RTDの温度(°C)。ファンクションは、OHMまたはOHMFでなければなりません。
FRTD85	22	結果 = αが0.00385の100Ω RTD(40654Aまたは406548)の温度(F)。ファンクションは、OHMまたはOHMFでなければなりません。
FRTD92	23	結果 = αが0.003916の100Ω RTDの温度(F)。ファンクションは、OHMまたはOHMFでなければなりません。

電源投入時の*operation_a,operation_b* = OFF,OFF
デフォルトの*operation_a,operation_b* = OFF,OFF

電源投入時のレジスタ値 = 次の例外を除き、すべてのレジスタが0に設定されます。

DEGREE = 20	REF = 1
SCALE = 1	RES = 50
PERC = 1	

備考

- FILTER、RMS、STATまたはPFAIL演算は、後続の読み取り値に対して実行されます。ただし、マルチメータの構成が変更されるたびに、前の演算結果は消去されて、新しい読み取り値に対して演算が繰り返されます。MATH OFFを設定する、MATHコマンドを実行して他の演算を指定する、またはポスト・プロセス演算をオンする(MMATHコマンドで説明してあるように、MMATH PFAILまたはMMATH STATは除く)まで、他のすべての演算はオンされたままとなります。

- 2つのリアルタイム演算がオンされている場合は、最初に*operation_a*が読み取り値に対して実行されます。次に、*operation_b*が最初の演算結果に対して実行されます。

- リアルタイム演算がオンされている場合、ディスプレイの1/2桁が0から9までフルの桁で表示されます。例えば、4.5桁のAC電圧測定を行った後、SCALE演算をオンすると、ディスプレイの5桁をフルに表示できます。

- 演算レジスタには、SMATHコマンドを使用して書き込みができます。また、演算レジスタは、RMATHコマンドを使用して読み取りができます。

- **問合せコマンド**　MATH?問合せコマンドは、オンされているリアルタイム演算機能を示す、カンマで区切られた2つの応答を返します。詳細については、本章の始めにある「問合せコマンド」を参照してください。

- **関連コマンド**: MMATH、RMATH、SMATH

例 次のプログラムは、20個の読み取り値に対してリアルタイムNULL演算を実行します。NULLコマンドの実行後、最初の読み取りが行50によりトリガされます。次に、OFFSETレジスタの値が3.05に変更されます。20回の読み取りが行90によりトリガされ、個々の読み取り値から3.05が差し引かれます。

```
10 OPTION BASE 1                    !配列の番号付けは1から始まります。
20 DIM Rdgs(20)                     !20個の読み取り値用の配列を宣言します。
30 OUTPUT 722;"PRESET NORM"         !プリセット、NRDGS 1、AUTO、DCV10
40 OUTPUT 722;"MATH NULL"           !リアルタイムNULL演算をオンにします。
50 OUTPUT 722;"TRIG SGL"            !1回の読み取りをトリガし、OFFSETに保存します。
60 OUTPUT 722;"SMATH OFFSET,3.05"   !3.05をOFFSETレジスタに書き込みます。
70 OUTPUT 722;"NRDGS 20"            !1回のトリガで20回の読み取り
80 OUTPUT 722;"TRIG SYN"            !SYNトリガ・イベント
90 ENTER 722;Rdgs(*)                !SYNイベント、NULL補正済み読み取り値の入力
100 PRINT Rdgs(*)                   !NULL補正済み読み取り値の出力
110 END
```

MCOUNT?

メモリ・カウントの問合せ　保存された読み取り値の総数を返します。

構文 **MCOUNT?**

備考
- 関連コマンド: MEM、MFORMAT、MSIZE、RMEM

例

```
10 OUTPUT 722; "MCOUNT?"     !保存された読み取り値の総数を返します。
20 ENTER 722;A               !応答を変数に入力します。
30 PRINT A                   !応答を出力します。
40 END
```

MEM

メモリ　読み取り値メモリをオンまたはオフにし、保存モードを指定します。

構文 **MEM [*mode*]**

mode

*mode*パラメータの選択肢は次のとおりです。

mode パラメータ	数値問合せの等価値	説明
OFF	0	読み取り値の保存を停止します(保存済みの読み取り値は元の状態のままです)。
LIFO	1	読み取り値メモリをクリアして、新しい読み取り値をLIFO(後入れ先出し)で保存します。

mode パラメータ	数値 問合せの 等価値	説明
FIFO	2	読み取り値メモリをクリアして、新しい読み取り値をFIFO (先入れ先出し)で保存します。
CONT	3	メモリをそのままの状態に保ち、前のモードを選択します (前のモードがない場合は、FIFOが選択されます)。

電源投入時の*mode* = OFF

デフォルトの*mode* = ON

備考

- 高速モードでは、FIFOモードでオンされている読み取り値メモリがいっぱいになった場合、トリガ・アーム・イベントがHOLDになり、これによって読み取りが停止して、マルチメータが高速モードから解除されます。読み取り値の一部またはすべてをメモリから削除すると、トリガ・アーム・イベントを変更することで(TARMコマンド)測定を再開できます。高速モードでなく、メモリがFIFOモードでいっぱいになった場合は、保存された読み取り値はそのままの状態で､新しい読み取り値は保存されません。LIFOモードでは、読み取り値メモリがいっぱいになると、高速モードかどうかに関係なく、最も古い読み取り値が最新の読み取り値に置換されます。

- コントローラがマルチメータのデータを要求して、その出力バッファが LIFO または FIFOモードで空の場合には、読み取り値がメモリから削除されて、コントローラへ送られます。これが、読み取り値の呼び出しの"暗黙の読み取り"法です。LIFOモードでは、最新の読み取り値が返されます。FIFOモードでは、最も古い読み取り値が返されます。読み取り値の保存モード(LIFOまたはFIFO)は、読み取り値の呼び出しとして"暗黙の読み取り"法を使用している場合にのみ重要です。読み取り値の保存モードは、RMEMコマンドを使用して呼び出された読み取り値には影響しません。

- MFORMAT コマンドを使用してメモリ・フォーマット (SINT、DINT、ASCII、SREAL またはDREAL)を指定します。

- RMEMコマンドを実行すると読み取り値メモリがOFFに設定されます。RMEMの実行後に読み取り値メモリの再度オンするには､MEM CONT､MEM FIFOまたはMEM LIFOを実行する必要があります。

- **問合せコマンド**　MEM? 問合せコマンドは、現在のメモリ・モードを返します。詳細については、本章の始めにある「問合せコマンド」を参照してください。

- **関連コマンド**: MCOUNT?、MFORMAT、MSIZE、RMEM

例

```
OUTPUT 722;"MEM FIFO"   !読み取り値メモリをFIFOモードでオンにします。
```

MENU

MENUコマンドは、前面パネルのアルファベット・コマンド・メニューのSHORTまたはFULLコマンド・リストを選択します。

構文 **MENU [*mode*]**

mode

*mode*パラメータには次に選択肢があります。

mode パラメータ	数値問合せの等価値	説明
SHORT	0	ショート・コマンド・メニューを選択します。
FULL	1	フル・コマンド・メニューを選択します。

電源投入時の*mode* = 電源が切断された場合に選択されるモード
デフォルトの*mode* = FULL

備考

備考
- アルファベット・コマンド・メニューにアクセスするには、シフト＋ MENU キー (C、E、L、N、R、SおよびTのラベルが付いた)のいずれかを押します。これで、上および下矢印キーを使用して、特定のコマンドを捜すことができます。
- *mode*パラメータは、不揮発性メモリに保存されます(電源を切断しても失われません)。
- FULLメニューには、コマンドにアクセスして疑問符を末尾につけることで作成できる問合せコマンド(例えばBEEP、BEEP?)以外のすべてのコマンドが含まれています。SHORTメニューは、GPIBバス関連コマンドと、前面パネル・キー専用コマンド(例えば、RSTATE コマンド、Recall Stateキー)を除いたものです。
- 問合せコマンド　MENU?問合せコマンドは、現在のメニュー・モードを示す応答を返します。詳細については、本章の始めにある「問合せコマンド」を参照してください。
- 関連コマンド: DEFKEY、LOCK

例
```
OUTPUT 722;"MENU SHORT"      !SHORTメニューを選択します。
```

MFORMAT

メモリ・フォーマット。読み取り値メモリをクリアして、新しい読み取り値の保存フォーマットを指定します。

構文 **MFORMAT [*format*]**

format

*format*パラメータの選択肢は次のとおりです。

format パラメータ	数値問合せ の等価値	説明
ASCII	1	読み取り値あたりASCII-16バイト*
SINT	2	単精度整数-16ビットの2の補数(読み取り値あたり2バイト)
DINT	3	倍精度整数-32ビットの2の補数(読み取り値あたり4バイト)
SREAL	4	単精度実数-(IEEE-754)32ビット(読み取り値あたり4バイト)
DREAL	5	倍精度実数-(IEEE-754)64ビット(読み取り値あたり8バイト)

* ASCIIフォーマットは、実際には読み取り値の15バイトとヌル文字の1バイトで、ヌル文字は保存されたASCII読み取り値の区切りにのみ使用されます。

電源投入時の*format* = SREAL
デフォルトの*format* = SREAL

備考

- マルチメータは、読み取り値ではなく値±1E+38をメモリに保存することで、過負荷を示します。過負荷値がディスプレイに呼び出された場合には、値±1E+38が表示されます。過負荷の値が読み取り値メモリからGPIB出力バッファに転送された場合は、指定された出力フォーマットの過負荷数値に変換されます(詳細はFORMATコマンドを参照してください)。

- SINTまたはDINTメモリ・フォーマットを使用している場合、マルチメータは、特定のスケール・ファクタを想定して個々の読み取り値を保存します。このスケール・ファクタは、現在の測定ファンクション、レンジ、A/D設定およびオンされている演算に基づきます。読み取り値を呼び出すと、マルチメータが現在の測定ファンクション、レンジ、A/D設定およびオンされている演算に基づいてスケール・ファクタを計算します。次に、スケール・ファクタに保存されている読み取り値を掛けて、結果(呼び出された読み取り値)をディスプレイまたは出力プリンタに送ります。したがって、データをSINTまたはDINTフォーマットで保存する場合と呼び出す場合のマルチメータの構成は、必ず同じにしてください。

- リアルタイムまたはポスト･プロセス演算機能がオンされている場合(STATまたはPFAILを除く)、あるいはオートレンジがオンされている場合は、周波数／周期測定で、SINTあるいはDINTの出力フォーマットまたはメモリ･フォーマットを使用しないでください。

- メモリ・フォーマットは、OFORMAT コマンドにより指定された出力フォーマットには影響しません。

- 読み取り値メモリは、MEMコマンドを使用してオンにします。保存されている読み取り値には、RMEMコマンドを使用するか、"暗黙の読み取り"によりアクセスします。"暗黙の読み取り"については、第4章の「読み取り値メモリの使用方法」で説明しています。

- サブサンプリング測定(SSACまたはSSDCコマンド)で読み取り値メモリを使用する場合は、メモリ・フォーマットはSINTに設定され、メモリ・モードはFIFO(MEM FIFOコマンド)に設定され、サンプルを取得する前に(MEM FIFOコマンドの実行により)読み取り値メモリが空でなければなりません。トリガ・アーム・イベントが発生したときにこれらの要件が満たされていなければ、エラーが生成されます。

- **問合せコマンド** MFORMAT?問合せコマンドは、現在のメモリ・フォーマットを返します。詳細については、本章の始めにある「問合せコマンド」を参照してください。

- **関連コマンド**: MCOUNT?、MEM、MSIZE、RMEM

例

```
10 OUTPUT 722;"NPLC 10"          !10 PLCSの積分時間
20 OUTPUT 722;"DCV 7"            !DC電圧、10Vレンジ
30 OUTPUT 722;"MATH OFF"         !演算機能をオフにします。
40 OUTPUT 722;"MEM FIFO"         !読み取り値メモリ(FIFOモード)をオンにします。
50 OUTPUT 722;"MFORMAT DINT"     !DINTメモリ・フォーマットを選択します。
60 END
```

保存されたデータを呼び出す場合は、マルチメータの構成がデータを保存したときと同じであることを確認してください。

MMATH

メモリ演算 後処理演算をオフまたはオンにします。

構文 **MMATH [*operation_a*] [,*operation_b*]**

operation

*operation*パラメータの選択肢は次のとおりです。

operation **パラメータ**	**等価数値**	**説明**
OFF	0	すべての後処理演算をオフにします。
CONT	1	前の演算をオンにします。2つの演算を再開するには、MMATH CONT,CONTを送ります。
CTHRM	3	結果 = 5kΩサーミスタ(40653B)の温度(°C)。ファンクションは、OHMまたはOHMF(10kΩレンジ以上)でなければなりません。
DB	4	結果 = $20 \times \log_{10}$(読み取り値/REFレジスタ)。REFレジスタが1に初期化され、dBVとなります。
DBM	5	結果 = $10 \times \log_{10}$(読み取り値^2/RES レジスタ/1mW). ファンクションは、ACV、DCVまたはACDCVでなければなりません。
FILTER	6	結果 = 指数重みデジタル・ローパス・フィルタの出力。応答は、DEGREEレジスタにより設定されます。

operation パラメータ	等価数値	説明
FTHRM	8	結果 = 5kΩサーミスタ(40653B)の温度(F)。ファンクションは、OHMまたはOHMF(10kΩレンジ以上)でなければなりません。
NULL	9	結果 = 読み取り値 – OFFSETレジスタ　OFFSETレジスタは、最初の読み取り値に設定されます。その後、これは変更できます。
PERC	10	結果 = ((読み取り値 – PERC レジスタ)/PERCレジスタ)×100
PFAIL	11	読み取り値対MAXおよびMINレジスタ
RMS	12	結果 = 読み取り値を2乗し、FILTER演算を適用して、平方根を求めます。
SCALE	13	結果 = (読み取り値 – OFFSET レジスタ)/SCALEレジスタ
STAT	14	現在の読み取り値グループに対して統計計算を実行して、結果を次のレジスタに保存します。 SDEV = 標準偏差 MEAN = 読み取り値の平均 NSAMP = 読み取り値の個数 UPPER = 最大読み取り値 LOWER = 最小読み取り値
CTHRM2K	16	結果 = 2kΩサーミスタ(40653A)の温度(°C)。ファンクションは、OHMまたはOHMFでなければなりません。
CTHRM10K	17	結果 = 10kΩサーミスタ(40653C)の温度(°C)。ファンクションは、OHMまたはOHMFでなければなりません。
FTHRM2K	18	結果 = 2kΩサーミスタ(40653A)の温度(F)。ファンクションは、OHMまたはOHMFでなければなりません。
FTHRM10K	19	結果 = 10kΩサーミスタ(40653C)の温度(F)。ファンクションは、OHMまたはOHMFでなければなりません。
CRTD85	20	結果= αが0.00385の100W RTD(40654Aまたは406548)の温度(°C)。ファンクションは、OHMまたはOHMFでなければなりません。
CRTD92	21	結果= αが0.003916の100W RTDの温度(°C)。ファンクションは、OHMまたはOHMFでなければなりません。
FRTD85	22	結果= αが0.00385の100W RTD(40654Aまたは406548)の温度(F)。ファンクションは、OHMまたはOHMFでなければなりません。
FRTD92	23	結果= αが0.003916の100W RTDの温度(F)。ファンクションは、OHMまたはOHMFでなければなりません。

電源投入時の***operation_a,operation_b*** = OFF,OFF

デフォルトの***operation_a,operation_b*** = OFF,OFF

電源投入時のレジスタ値 = 次の例外を除き、すべてのレジスタが0に設定されます。

DEGREE = 20	REF = 1
SCALE = 1	RES = 50
PERC = 1	

備考

- オンされた後処理演算は、STATとPFAILを除いて、読み取り値が削除される際、または読み取り値メモリからディスプレイまたはGPIB出力バッファに移動またはコピーされる際に、各読み取り値に対して実行されます(メモリ内の読み取り値は、後処理演算により変更されません)。STATまたはPFAIL後処理演算は、MMATHコマンドの実行直後に、メモリ内の読み取り値を使用して実行されます(MMATHコマンドが実行されると、STATおよびPFAIL演算は、メモリ内に追加された読み取り値に対して更新されません)。

- STAT演算の場合、結果はSDEV、MEAN、NSAMP、UPPERおよびLOWER演算レジスタに保存されます(これらのレジスタについては、RMATHコマンドを参照してください)。

- PFAIL演算の場合、リミット外の読み取り値が検出されると、常にステータス・レジスタのビット番号1が設定され（これにより、RQSコマンドによりオンされていれば、GPIB SRQがfineに設定されます)、ディスプレイにはFAILED LOWまたはFAILED HIGHメッセージが表示されます。

- オンされている後処理演算は、MMATH OFFを設定する、リアルタイム演算(MATHコマンド)をオンにする、またはMMATHコマンドを実行して別の演算を指定する(以下の備考で説明する例外を除く)まで、オンされたままとなります。

- 前面パネルからMMATHを実行した場合、結果はディスプレイのみに送られます。MMATHがリモートから実行された場合、結果は出力バッファのみに送られます。

- 2つの後処理演算がオンされている場合、最初に*operation_a*が読み取り値に対して実行されます。次に、*operation_b*が最初の演算結果に対して実行されます。

- 後処理演算がオンされている場合、ディスプレイの1/2桁が0から9までフルの桁で表示されます。例えば、4.5桁のAC電圧測定を行っている場合、SCALE演算をオンすると、ディスプレイの5桁をフルに表示できます。

- 演算レジスタには、SMATHコマンドを使用して書き込みができます。また、演算レジスタは、RMATHコマンドを使用して読み取りができます。

- **問合せコマンド**　MMATH?問合せコマンドは、現在オンされている後処理演算機能を示す2つの応答(カンマで区切られています)を返します。

- RMEMコマンドを使用して読み取り値を呼び出すと、読み取り値メモリがオフされます。これは、新しい読み取り値が読み取り値メモリに入れられず、オンされたメモリ演算をこれらの値に対して実行できないということです。"暗黙の読み取り"法を使用して読み取り値を呼び出した場合、読み取り値メモリはオフされません。

- **関連コマンド**: MATH、MEM、RMATH、RMEM、SMATH

例 次のプログラムは、20個の読み取り値に対して後処理NULL演算を実行します。MMATH NULLコマンドの実行後、21個の読み取り値が取得され、読み取り値メモリにFIFOモードで保存されます。行80が、OFFSETレジスタに保存されている最初の読み取り値を呼び出します。次に、OFFSETレジスタ内の値が3.05に変更されます。メモリ内の残りの20個の読み取り値が呼び出され、それぞれに対してNULL演算が実行されます。

```
10 OPTION BASE 1                     !配列の番号付けは1から始まります。
20 DIM Rdgs(20)                      !配列のサイズを読み取り値20個に設定します。
30 OUTPUT 722; "PRESET NORM"         !プリセット、NRDGS 1、AUTO、DCV10
40 OUTPUT 722;"MEM FIFO"             !読み取り値メモリをFIFOモードでオンにします。
50 OUTPUT 722;"MMATH NULL"           !後処理NULL演算をオンにします。
60 OUTPUT 722;"NRDGS 21"             !1回のトリガで21個の読み取り
70 OUTPUT 722;"TRIG SGL"             !読み取りをトリガします。
80 ENTER 722;A                       !暗黙の読み取りを使用して最初の読み取り値を呼び出します。
90 OUTPUT 722; "SMATH OFFSET, 3.05"!3.05をオフセット・レジスタに書き込みます。
100 ENTER 722;Rdgs(*)                !暗黙の読み取りを使用して読み取り値を呼び出します。
105                                  !個々に対してNULL演算を実行します。
110 PRINT Rdgs(*)                    !NULL演算した読み取り値を出力します。
120 END
```

MSIZE

メモリ・サイズ　以前のマルチメータでは、MSIZEコマンドは、すべてのメモリをクリアして、読み取り値、サブプログラムおよびステート保存用のメモリ・スペースを割り当てるために使用されていました。3458は、言語の互換性を維持するためにMSIZEコマンドを受け付けますが、3458のメモリ割り当ては定義済みで変更できないため、何のアクションも実行されません。ただし、MSIZE?問合せコマンドは、読み取り値メモリの合計量とサブプログラム/ステート・メモリの未使用最大ブロックの確認に役立ちます。

構文 **MSIZE [*reading_memory*][,*subprogam_memory*]**

備考
- サブプログラム/ステート・メモリは、使用するにつれて最終的に断片化して、多数の小さなブロックになってしまいます。MSIZE?コマンドは、読み取り値メモリの合計バイト数とサブプログラム/ステート・メモリの最大未使用ブロックのバイト数を返します。SCRATCHコマンドは、すべてのサブプログラムとステートをメモリからクリアして、メモリ領域を1つの連続したブロックに戻します。また、電源を入れなおすと、可能な限り、メモリの断片化したブロックを結合します。

- **問合せコマンド**　MSIZE? 問合せコマンドはカンマで区切られた 2 つの応答を返します。1番目の応答は、読み取り値メモリの合計バイト数です。2番目の応答は、未使用サブプログラム/ステート・メモリの最大ブロック(単位バイト)です。

- **関連コマンド**: MCOUNT?、MEM、MFORMAT、RMEM、DELSUB、SCRATCH、SUB、SUBEND、SSTATE

例
```
10 OUTPUT 722; "MSIZE?"      !メモリ・サイズを問い合わせます。
20 ENTER 722;A,B             !応答を入力します。
30 PRINT A,B                 !応答を出力します。
40 END
```

NDIG

桁数　マルチメータが表示する桁数を指定します。

構文 **NDIG [*value*]**

value

*value*パラメータは3～8の整数です(暗黙の1/2桁があります。つまり、NDIG 3を指定した場合、マルチメータは3 1/2桁を表示します)。

電源投入時の*value* = 7 (7 1/2桁)
デフォルトの*value* = 7 (7 1/2桁)

備考
- NDIGコマンドは、表示される最大桁数を設定します。これは、A/Dコンバータの分解能、またはメモリあるいはGPIBバスに送られる読み取り値には影響しません。マルチメータは、A/Dコンバータで分解できる以上の桁数は表示できません。

- **問合せコマンド**　NDIG? 問合せコマンドは、現在指定されている桁数を返します。詳細については、本章の始めにある「問合せコマンド」を参照してください。

- **関連コマンド**: DISP

例
```
10 OUTPUT 722; "RESET"    !電源投入時のステートに戻ります。
20 OUTPUT 722;"NDIG 8"    !8 1/2桁を表示します。
30 END
```

NPLC

電源周波数。A/Dコンバータの積分時間を電源ライン周波数で指定します。積分時間は、A/Dコンバータが入力信号を測定する時間です。

構文 NPLC [*power_line_cycles*]

power_line_cycles

NPLCコマンドは、主にA/Dコンバータの基準周波数(LFREQコマンド)におけるノーマル・モード・ノイズ除去(NMR)の設定に使用します。*power_line_cycles*パラメータの値が≧1の場合は、電源ライン周波数におけるNMRは少なくても60dBとなります。値が＜1の場合には、NMRはありません。A/Dコンバータの積分時間が設定されるだけです。*power_line_cycles*パラメータの範囲と増分ステップ・サイズは次のとおりです。

基準周波数が60Hz(LFREQコマンド)の場合、0〜1PLC、0.000006PLCステップ

または

基準周波数が50Hzの場合、0〜1PLC、0.000005PLCステップ

1〜10PLC 、ステップ1PLC

10〜1000PLC 、ステップ10PLC

電源投入時の*power_line_cycles* = 10
デフォルトの*power_line_cycles* = 0(最小積分時間500nsが選択されます)

積分時間(PLCで表します)、A/D コンバータの基準周波数(LFREQ コマンド)および分解能桁数の関係は次のとおりです。

分解能桁数			電源ライン周波数(NPLCコマンド)	
DCV	DCI、OHM (F)	ACI、ACDCI、ACV*、ACDCV*	基準周波数(LFREQ) = 60HZ	基準周波数(LFREQ) = 50HZ
4.5	4.5	4.5	0〜0.000030	0〜0.000025
5.5	5.5	5.5	0.000036〜0.000360	0.000030〜0.000300
6.5	6.5	6.5	0.000366〜0.030000	0.000305〜0.025000
7.5	7.5**	6.5	0.030006〜1	0.025005〜1
8.5**	7.5**	6.5	2〜1000	2〜1000

*アナログ測定法のみ(SETACV ANAコマンド)

** 10Ω OHM(F)レンジと100mV DCVレンジを除くすべてのレンジ。10Ω OHM(F)レンジは最大6.5桁、100mV DCVレンジは最大7.5桁です。

備考

- ACVおよびACDCV(SETACV ANA法のみ)、ACT、ACDCI、DCI、DCV、OHM、OHMFの測定ファンクションでは、A/Dコンバータの積分時間で分解能が決まります。積分時間は、FREQとPERには影響しません。サンプリングしたACVまたはACDCV(SETACV SYNCまたはSETACV RNDM)では、積分時間は自動的に選択され、サンプル数を変化させることで指定した分解能が実現されます。直接サンプリングしたデジタイズまたはサブサンプリングしたデジタイズでは、積分時間は固定されており、変更できません。

- NPLCコマンドとAPERコマンドはどちらも積分時間を設定するため、一方を実行すると、もう一方によって設定済みの積分時間はキャンセルされます。RESコマンド、またはファンクション・コマンドあるいはRANGEコマンドの*%_resolution*パラメータを使用しても、間接的に積分時間を選択できます。分解能を指定する場合、NPLC(またはAPER)との関係で次のようになります。

- 分解能を指定する**前に**NPLC(またはAPER)コマンドを送ると、マルチメータは大きな方の分解能(長い方の積分時間)を指定するコマンドに従います。

- 分解能を指定した**後に**NPLC(またはAPER)コマンドを送ると、そのNPLC(またはAPER)コマンドにより指定された積分時間が使用され、前に指定済みの分解能は無視されます。

- より一般的なやり方は、上の2つの最初の方です。つまり、最初にNPLCコマンドを実行してノーマル・モード・ノイズ除去(NMR)を設定し、次にファンクション・コマンドまたはRANGEコマンドを使用して*%_resolution*を指定します。これで、NMRと最低限必要な分解能が確実に得られます。

- **問合せコマンド**　NPLC?問合せコマンドは、A/Dコンバータが使用する積分時間(PLC単位)を返します。積分時間は、APER、NPLCあるいはRESコマンド、またはファンクション・コマンドあるいはRANGEコマンドの*%_resolution*パラメータによって指定できるため、NPLCによる最終の指定と異なるPLC数をNPLC?コマンドが返す場合があります。

- **関連コマンド**: APER、FUNC、LFREQ、RE

例　次のプログラムでは、行10でPLC数を最小に設定して、行20による分解能の設定を可能にしています。行20が指定する分解能は100µVです。

```
10 OUTPUT 722;"NPLC 0"          !PLC数を最小に設定します。
20 OUTPUT 722;"DCV 6,.00167"    !DC電圧、最大6V、分解能100µV
30 END
```

次のプログラムでは、行10でPCLを1000に設定しています。これは、最大分解能に相当し、このために行20の*%_resolution*は測定に影響しません。行20で要求されている分解能は10mΩですが、行10のために実際の分解能は100µΩになります。

```
10 OUTPUT 722;"NPLC 1000"       !PLC数を最大に設定します。
20 OUTPUT 722;"OHM 1E3,.001"    !2端子抵抗、1kΩ最大入力を選択します。
30 END
```

NRDGS

読み取り値の回数。1回のトリガで取得される読み取り回数と、個々の読み取りを開始するイベント(サンプル・イベント)を指定します。

構文　**NRDGS [*count*][,*event*]**

count

トリガ・イベントあたりの読み取り回数を指定します。このパラメータの有効範囲は、1〜16777215です(*count*パラメータは、RMEMコマンドのrecordパラメータにも相当します。詳細は、RMEMコマンドを参照してください)。

電源投入時の*count* = 1
デフォルトの*count* = 1

event

個々の読み取りを開始するイベント(サンプル・イベント)を指定します。*event*パラメータは次のとおりです。

***event* パラメータ**	**数値 問合せの 等価値**	**説明**
AUTO	1	マルチメータがビジーでない場合は、常に読み取りを開始します。
EXTSYN	2	マルチメータの外部トリガ入力コネクタの立ち下がりエッジで読み取りを開始します。
SYN	5	マルチメータの出力バッファが空、読み取り値メモリがオフまたは空、かつコントローラがデータを要求した場合に読み取りを開始します。
TIMER*	6	AUTOに似ており、連続した読み取りの間の時間間隔があります(間隔はTIMERコマンドで指定します)
LEVEL**	7	SLOPEコマンドにより指定されたスロープ上の、LEVELコマンドにより指定された電圧に入力信号が達すると読み取りを開始します。
LINE*	8	ACライン電圧のゼロ交差で読み取りを開始します。

* TIMERイベントまたはLINEイベントは、サンプリングしたACあるいはAC+DC電圧の測定(SETACV RNDMあるいはSYNC)、または周波数／周期測定には使用できません。

** LEVELサンプル・イベントは、DC電圧測定とサンプリングした測定でのみ使用できます。

電源投入時の*event* = AUTO
デフォルトの*event* = AUTO

備考

- TIMERイベントは読み取り間の間隔を指定するため、**カウント**が1より大きい場合にのみ適用されます。最初の読み取りは、TIMER間隔を置かずに行われます。ただし、DELAYコマンドを使用して、最初の読み取りの前に時間間隔を挿入できます(TIMERイベントは、オートレンジングを中断します)。

- SWEEPコマンドを使用すれば、2つのコマンド、NRDGS *n*,TIMERとTIMER *n*を置換することができます。SWEEPコマンドは、読み取り回数と読み取り間の間隔を指定します。これらのコマンドには互換性があります。マルチメータは、プログラミングの中で最後に実行されたコマンドを使用します。SWEEPコマンドを実行すると、サンプル・イベントをTIMERに自動的に設定されます。電源投入時、RESETまたはPRESETのステートでは、NRDGSコマンドが使用されます。

- SYN を複数のトリガ・アーム・イベント、トリガ・イベントまたはサンプル・イベントに対して使用した場合は、SYNイベントの1回の発生で、指定されたSYNイベント要件のすべてが満たされます。これを、下の2番目の"SYNイベント"の例に示します。

- **問合せコマンド**　NRDGS?問合せコマンドは、カンマで区切られた2つの応答を返します。1番目の応答は、トリガあたりの読み取り値の個数です。2番目の応答は、現在のサンプル・イベントです。詳細については、本章の始めにある「問合せコマンド」を参照してください。

- **関連コマンド**: DELAY、LEVEL、RMEM、SLOPE、TARM、TIMER、TRIG、SWEEP

例 SYNイベント

次のプログラムでは、行70がマルチメータのデータを要求します。これが、SYNイベントを満たし、読み取りが開始されます。次に、読み取り値がコントローラへ送られ、出力されます。このプロセスが、3つの読み取り値が取得されて、出力されるまで繰り返されます。

```
10 OPTION BASE 1                !配列の番号付けは1から始まります。
20 DIM A(3)                     !配列を宣言します。
30 OUTPUT 722;"DCV 8,.00125"    !DC電圧、10Vレンジ、分解能100μV
40 OUTPUT 722;"NRDGS 3, SYN"    !3個の読み取り値/トリガ、SYNサンプル・イベント
50 OUTPUT 722;"TRIG AUTO"       !AUTOトリガ・モード
60 ENTER 722;A(*)               !読み取りを入力します。
70 PRINT A(*)                   !読み取り値を出力します。
80 END
```

次のプログラムでは、SYNがトリガ・アーム・イベント、トリガ・イベントおよびサンプル・イベントに対して指定されます。1回のトリガあたり5個の読み取り値が指定されています。SYNイベント(行60)の1回の発生で、トリガ・アーム・イベント、トリガ・イベントおよびサンプル・イベントが満たされ、最初の読み取り値が開始されます。残りの4つの読み取りを開始するには、さらに4つのSYNイベント(個々の読み取りに対して1つ)が必要です。

```
10 OPTION BASE 1                !配列の番号付けは1から始まります。
20 DIM Rdgs(5)                  !読み取り値用の配列を宣言します。
30 OUTPUT 722;"PRESET NORM"     !SYNトリガ・イベント、DCV、NPLC 1、MEM OFF
40 OUTPUT 722;"TARM SYN"        !SYNトリガ・アーム・イベント
50 OUTPUT 722;"NRDGS 5, SYN"    !5個の読み取り値/トリガ、SYNサンプル・イベント
60 ENTER 722;Rdgs(*)            !SYNイベント、読み取り値を入力します。
70 PRINT Rdgs(*)                !読み取り値を出力します。
80 END
```

TIMER

次のプログラムでは、同期トリガ(行60)に応答して4個の読み取り値が取得されます。1個めの読み取りは、あらかじめプログラムされたデフォルトの遅延の直後に行われます。残りの3個はそれぞれ200ms間隔で行われます。

```
10 OPTION BASE 1             !配列の番号付けは1から始まります。
20 DIM Rdgs(4)               !読み取り値用の配列のサイズを宣言します。
30 OUTPUT 722;"PRESET NORM"    !TARM AUTO、TRIG SYN、DCVオートレンジ
40 OUTPUT 722;"TIMER 200E-3"!タイマ間隔を200m秒に設定します。
```

```
50 OUTPUT 722;"NRDGS 4,TIMER"  !4回の読み取り/トリガおよびタイマを選択します。
60 ENTER 722;Rdgs(*)           !読み取り値をトリガして入力します。
70 PRINT Rdgs(*)               !読み取り値を出力
80 END
```

OCOMP

OCOMPコマンドは、オフセット補正抵抗ファンクションをオンまたはオフにします。

構文 **OCOMP [*control*]**

control

*control*パラメータの選択肢は次のとおりです。

control パラメータ	数値 問合せの 等価値	説明
OFF	0	オフセット補正抵抗がオフされます。
ON	1	オフセット補正抵抗がオンされます。

電源投入時の*control* = OFF
デフォルトの*control* = ON

備考
- オフセット補正がオンされている場合、個々の抵抗読み取りの前に外部オフセット電圧が測定され(抵抗電流源はオフ)、このオフセットが次の読み取り値から差し引かれます。これにより、オフセット電圧が抵抗測定に影響するのを防ぎますが、1回の読み取りに必要な時間は2倍になります。

- オフセット補正抵抗は、2端子および4端子抵抗測定のどちらでも使用できます。オフセット補正をオンにして、抵抗から他の測定ファンクション(DCV、ACVなど)に変更した場合、オフセット補正は一時的にオフされます。ただし、2端子または4端子抵抗に戻ると、オフセット補正は再度オンされます。

- オフセット補正は、10Ω～100kΩレンジでのみ実行可能です。1MΩ～1GΩレンジを使用する場合にOCOMPがオンされると、オフセット補正なしで読み取りが行われます。

- **問合せコマンド**　OCOMP? 問合せコマンドは、現在のオフセット補正モードを返します。詳細については、本章の始めにある「問合せコマンド」を参照してください。

- **関連コマンド**: OHM、OHMF

例
```
OUTPUT 722;"OCOMP ON"        !オフセット補正をオンにします。
```

OFORMAT

出力フォーマット。直接コントローラに送られる読み取り値、または読み取り値メモリからコントローラへ転送される読み取り値のGPIB出力フォーマットを指定します。

構文 **OFORMAT [*format*]**

format

*format*パラメータの選択肢は次のとおりです。

format **パラメータ**	**数値問合せの等価値**	**説明**
ASCII	1	読み取り値あたりASCII-15バイト(下の1、2番目の備考を参照)
SINT	2	単精度整数-16ビットの2の補数(読み取り値あたり2バイト)
DINT	3	倍精度整数-32ビットの2の補数(読み取り値あたり4バイト)
SREAL	4	単精度実数-(IEEE-754)32ビット(読み取り値あたり4バイト)
DREAL	5	倍精度実数-(IEEE-754)64ビット(読み取り値あたり8バイト)

電源投入時の*format* = ASCII
デフォルトの*format* = ASCII

備考

- ASCII出力フォーマットでは、ほとんどのコンピュータへの転送の終わりを示すために*cr lf*(キャリッジ・リターン、改行)が送られます。ただし、SINT、DINT、SREALおよびDREAL出力フォーマットでは、*cr lf*は送られません。どのフォーマットでも、ENDコマンドによりGPIB EOI機能を使用して転送の終わりを示すことができます。詳細は、ENDコマンドを参照してください。

- ASCIIフォーマットを使用する場合は、キャリッジリターン、改行(*cr,lf*)の行末シーケンス用の2バイトが余分に必要です。*cr,lf*はASCIIフォーマットの場合にのみ使用され、通常はASCIIフォーマットの個々の読み取り値の後に続きます。ただし、ASCII出力フォーマットを使用していて、複数の読み取り値がRMEMを使用して読み取り値メモリから呼び出された場合は、マルチメータが読み取り値の間にカンマを入れます(カンマ = 1 バイト)。この場合、*cr,lf*は、呼び出されるグループの最後の読み取り値の後に1回だけ現れます。読み取り値が直接バスに出力される場合(読取り値メモリがオフ)、読み取り値が"暗黙の読み取り"を使用して呼び出される場合、または他の出力フォーマットを使用している場合には、カンマは使用されません。

- マルチメータは、次のように、特定の出力フォーマットに対して、可能な最大の数値を出力することで過負荷状態(入力が現在のレンジで測定可能な範囲を超えている)を示します。

SINTフォーマット: +32767または-32768(スケーリングなし)

DINTフォーマット: +2.147483647E+9または-2.147483648E+9 (スケーリングなし)

ASCII、SREAL、DREAL: ±1.OE+38

- 読み取り値メモリがオフの場合は、前に指定したフォーマットに関係なく、SSACコマンドまたはSSDCコマンド(サブサンプリング)により出力フォーマットがSINTに自動的に設定されます。サブサンプリングを行い、読み取り値メモリを使用しない場合は、SINT出力フォーマットを使用する必要があります。

- 出力フォーマットは、GPIBバスを使用して転送される読み取り値のみが対象です。指定された出力フォーマットに関係なく、問合せコマンドに対する応答は常にASCIIフォーマットです。問合せ応答の後、出力フォーマットは指定したタイプに戻ります。出力フォーマットは、MFORMATコマンドにより指定されたメモリ・フォーマットには影響しません。

- SINTまたはDINT出力フォーマットを使用すると、スケール・ファクタが個々の読み取り値に適用されます。スケール・ファクタは、現在の測定ファンクション、レンジ、A/D設定およびオンされた演算に基づきます。したがって、スケール・ファクタを取得(ISCALE?コマンド)するときのマルチメータの校正が、読み取りが行われた際と同じになるようにしてください。

- リアルタイムあるいは後処理演算がオンの場合(STATあるいはPFAILをのぞく)、またはオートレンジがオンの場合には、周波数／周期測定にSINTあるいはDINT出力/メモリ・フォーマットを使用しないでください。

- **問合せコマンド**　OFORMAT?　問合せコマンドは、現在の出力フォーマットを返します。詳細については、本章の始めにある「問合せコマンド」を参照してください。

- **関連コマンド**: END、ISCALE?、MFORMAT、QFORMAT

例　SINTフォーマット

次のプログラムは、10個の読み取り値をSINTフォーマットで出力し、スケール・ファクタを取得して、スケール・ファクタを個々の読み取り値に掛けます。

```
10 OPTION BASE 1                    !配列の番号付けは1から始まります。
20 INTEGER Num_readings             !変数を宣言します。
30 INTEGER Int_rdgs (1:10) BUFFER !整数バッファ配列を作成します。
40 REAL Rdgs(1:10)                  !実数配列を作成します。
50 Num_readings=10                  !読み取り値の個数 = 10
60 ASSIGN @Dvm TO 722               !マルチメータ・アドレスを割り当てます。
70 ASSIGN @Int_rdgs TO BUFFER Int_rdgs(*)!バッファ I/Oパス名を割り当てます。
80 OUTPUT @Dvm;"PRESET NORM;OFORMAT SINT;NPLC 0;NRDGS ";Num_readings
85 !TARM AUTO、TRIG SYN、SINT 出力フォーマット、最小積分時間
90 TRANSFER @Dvm TO @Int_rdgs;WAIT!SYNイベント、読み取り値を整数配列に
91 !変換します。コンピュータの整数フォーマットがSINTと同じため、
95 !ここでは、データ変換は不要です(整数配列は必要)。
100 OUTPUT @Dvm;"ISCALE?"          !フォーマットのスケール・ファクタを問い合わせます。
110 ENTER @Dvm;S                    !スケール・ファクタを入力します。
```

```
120 FOR I=1 TO Num_readings
130 Rdgs(I)=Int_rdgs(I)            !個々の整数読み取り値を実数フォーマットに転送
135 !します(次の行での整数のオーバフローを防ぐために必要)。
140 R=ABS(Rdgs(I))                 !絶対値を使用してOVLDがないか確認します。
150 IF R>=32767 THEN PRINT "OVLD" !OVLDの場合、過負荷メッセージを出力します。
160 Rdgs(I)=Rdgs(I)*S              !読み取り値にスケール・ファクタを掛けます。
170 Rdgs(I)=DROUND(Rdgs(I),4)      !4桁に丸めます。
180 NEXT I
190 END
```

DINTフォーマット

次のプログラムは前のプログラムと似ていますが、50回の読み取りを行って、これら読み取り値をDINフォーマットを使用してコンピュータへ転送する点が異なります。

```
10 OPTION BASE 1                   !配列の番号付けは1から始まります。
20 INTEGER Num_readings,I,J,K      !変数を宣言します。
30 Num_readings=50                 !読み取り値の個数 = 50
40 ALLOCATE REAL Rdgs(1:Num_readings)!読み取り値用の配列を作成します。
50 ASSIGN @Dvm TO 722              !マルチメータ・アドレスを割り当てます。
60 ASSIGN dBuffer TO BUFFER[4*Num_readings] !バッファ I/Oパス名を割り当てます。
70 OUTPUT @Dvm;"PRESET NORM;RANGE 10;OFORMAT DINT;NRDGS ";Num_readings
75 !TARM AUTO、TRIG SYN、DCV 10Vレンジ、DINT出力フォーマット、NRDGS 50、AUTO
80 TRANSFER @Dvm TO @Buffer;WAIT  !SYNイベント、読み取り値の転送
90 OUTPUT @Dvm; "ISCALE?"          !DINTのスケールを問い合わせます。
100 ENTER @Dvm; S                  !スケール・ファクタを入力します。
110 FOR I=1 TO Num_readings
120 ENTER @Buffer USING "#,W,W";J,K!1つの16ビットの2の補数ワードを
121 !各変数JおよびKに入力します(# = 文ターミネータは
125 !不要です。W = データを、16ビットの2の補数として入力します)。
130 Rdgs(I)=(J*65536.+K+65536.*(K<O))!実数に変換します。
140 R=ABS(Rdgs(I))                 !絶対値を使用してOVLDがないか確認します。
150 IF R>2147483647 THEN PRINT "OVLD"!過負荷が発生した場合、メッセージを出力します。
160 Rdgs(I)=Rdgs(I)*S              !スケール・ファクタを適用します。
170 Rdgs(I)=DROUND(Rdgs(I),8)      !変換された読み取り値を丸めます。
180 PRINT Rdgs(I)                  !読み取り値を出力します。
190 NEXT I
200 END
```

SREALフォーマット

次のプログラムは、SREALフォーマットの10個の読み取り値出力を変換する方法を示します。

```
10 OPTION BASE 1                   !コンピュータの番号付けが1から始まります。
20 INTEGER Num_readings            !変数を宣言します。
30 Num_readings=10                 !読み取り値の個数 = 10
40 ALLOCATE REAL Rdgs(1:Num_readings)!読み取り値用の配列を作成します。
50 ASSIGN @Dvm TO 722              !マルチメータ・アドレスを割り当てます。
60 ASSIGN @Buffer TO BUFFER [4*Num_readings]!バッファ I/Oパス名を割り当てます。
```

```
70 OUTPUT @Dvm;"PRESET NORM;OFORMAT SREAL;NRDGS ";Num_readings
75 !TRIG SYN、SREAL出力フォーマット、1 PLC、DCVオートレンジ、10読み取り値
80 TRANSFER @Dvm TO @Buffer;WAIT  !SYNイベント、読み取り値の転送
90 FOR I=1 TO Num_readings
100 ENTER @Buffer USING "#,B";A,B,C,D!1つの8ビット・バイトを各変数に
101 !入力します(#=文ターミネータは不要です。B=1つの8ビット・バイトを
105 !入力して、0〜255の整数として解釈します)。
110 S=1                              !読み取り値をSREALから変換します。
120 IF A>127 THEN S=-1               !読み取り値をSREALから変換します。
130 IF A>127 THEN A=A-128            !読み取り値をSREALから変換します。
140 A=A*2- 127                       !読み取り値をSREALから変換します。
150 IF B>127 THEN A=A+l              !読み取り値をSREALから変換します。
160 IF B<=127 THEN B=B+128           !読み取り値をSREALから変換します。
170 Rdgs(I)=S*(B*65536.+C*256.+D)*2^(A-23)!読み取り値をSREALから変換します。
180 Rdgs(I)=DROUND(Rdgs(I),7)        !読み取り値を7桁に丸めます。これは、
181 !OVLD値が確実に1.E+38に丸められるように、SREALを使用して行う必要が
185 !あります(丸めない場合、値はやや小さくなります)。
190 IF ABS(Rdgs(I)=1.E+38 THEN       !過負荷が発生した場合:
200 PRINT "Overload Occurred"        !過負荷メッセージを出力します。
210 ELSE                             !過負荷がない場合:
220 PRINT Rdgs(I)                    !読み取り値を出力します。
230 END IF
240 NEXT I
250 END
```

DREALフォーマット

次のプログラムでは、DREAL出力フォーマットを使用しています。このフォーマットを使用する場合、変換は不要です。これは、DREALが、コントローラがその内部データ・フォーマット(8バイト/ワード)として使用するフォーマットと同じためです。

```
10 OPTION BASE 1                 !配列の番号付けは1から始まります。
20 REAL Rdgs(1:10) BUFFER        !バッファ配列を作成します。
30 ASSIGN @Dvm TO 722            !マルチメータ・アドレスを割り当てます。
40 ASSIGN @Rdgs TO BUFFER Rdgs(*) !バッファ I/Oパス名を割り当てます。
50 OUTPUT @Dvm;"PRESET NORM;NPLC 10;OFORMAT DREAL;NRDGS 10"
55 !TRIG SYN、10 PLC、DCVオートレンジ、DREAL出力フォーマット、10 RDGS/TRIG
60 TRANSFER @Dvm TO @Rdgs;WAIT!SYNイベント、読み取り値の転送
70 FOR I=1 TO 10
80 IF ABS(Rdgs(I))=l.E+38 THEN !過負荷が発生した場合:
90 PRINT "OVERLOAD OCCURRED"    !過負荷メッセージを出力します。
100 ELSE                         !過負荷がない場合:
110 Rdgs(I)=DROUND(Rdgs(I),8)   !読み取り値を丸めます。
120 PRINT Rdgs(I)               !読み取り値を出力します。
130 END IF
140 NEXT I
150 END
```

前のプログラムでは、TRANSFER文を使用してマルチメータから読み取り値を取得していました。次のプログラムでは、ENTER文を使用して、DREALフォーマットで読み取り値をコンピュータに転送します。ENTER文はI/Oパスが不要なためより使いやすいものの、TRANSFER文に比べると非常に遅くなります。また、ENTER文を使用する場合は、FORMAT OFFコマンドを使用して、コントローラにASCIIではなくその内部データ構造を使用するように命令する必要があります。

```
10 OPTION BASE 1                    !配列の番号付けは1から始まります。
20 Num_readings=20                  !読み取り値の個数 = 20
30 ALLOCATE REAL Rdgs(l:Num_readings)!読み取り値用の配列を作成します。
40 ASSIGN @Dvm TO 722               !マルチメータ・アドレスを割り当てます。
50 OUTPUT @Dvm;"PRESET NORM;OFORMAT DREAL;NPLC 10;NRDGS ";Num_readings
55 !TRIG SYN、DCVオートレンジ、DREAL出力フォーマット、10 PLC、20個の読み取り値/トリガ
60 ASSIGN @Dvm;FORMAT OFF           !8バイト/ワード・データ構造を使用します。
70 FOR I=1 TO Num_readings
80 ENTER @Dvm;Rdgs(I)               !個々の読み取り値を入力しします。
90 IF ABS(Rdgs(I))=1.E+38 THEN      !過負荷が発生した場合:
100 PRINT "OVERLOAD OCCURRED"       !過負荷メッセージを出力します。
110 ELSE                            !過負荷がない場合
120 Rdgs(I)=DROUND(Rdgs(I),8)       !読み取り値を丸めます。TO 8 DIGITS
130 PRINT Rdgs(I)                   !読み取り値を出力します。
140 END IF
150 NEXT I
160 END
```

OHM、OHMF

FUNCコマンドを参照してください。

OPT?

オプションの問合せ。マルチメータにインストールされたオプションを示す応答を返します。可能な応答は次のとおりです。

0 = インストールされたオプションなし。
1 = 拡張読み取り値メモリ・オプション

構文 **OPT?**

備考
- 関連コマンド: QFORMAT

例

```
10 OUTPUT 722;"OPT?"          !インストールされているオプションを問い合わせます。
20 ENTER 722;A$               !応答を入力します。
30 PRINT A$                   !応答を出力します。
40 END
```

PAUSE

サブプログラムの実行を中断します。中断したサブプログラムは、CONTコマンドを使用するか、GPIB Group Execute Triggerコマンドを実行することで再開できます。

構文 **PAUSE**

備考

- PAUSEコマンドは、サブプログラム内でのみ使用できます。
- 中断ステートに保たれるサブプログラムは1つだけです。あるサブプログラムの中断中に、実行されている別のサブプログラムが中断されると、最初のサブプログラムは終了し、2番目のサブプログラムは中断されたままになります。
- 入力バッファがオフの場合(INBUF OFFコマンド)、通常GPIBバスは、呼び出されたサブプログラムが完全に実行されるまでマルチメータによりホールドされます。サブプログラム内でPAUSEコマンドが現れると、GPIBバスが直ちに開放されます。
- ネストされたPAUSEコマンドは使用できません。つまり、サブプログラムが別のサブプログラムから呼び出される場合、呼び出されるサブプログラムの中にPAUSEコマンドを含めることはできません。
- **問合せコマンド**　PAUSE?問合せコマンドは、サブプログラムが現在中断されているかどうかを示す応答を返します。可能な応答は、サブプログラムが中断されていることを示すYES(数値問合せの等価値 = 1)、またはNO(数値問合せの等価値 = 0)。
- **関連コマンド**: CALL、COMPRESS、CONT、DELSUB、TRIGGER (GPIB コマンド)、SCRATCH、SUB、SUBEND

例

```
10 OUTPUT 722;"SUB OHMAC1"      !OHMAC1という名前のサブプログラムを保存します。
20 OUTPUT 722;"PRESET NORM"     !トリガを中断、プリセット
30 OUTPUT 722;"MEM FIFO"        !読み取り値メモリをFIFOモードでオンにします。
40 OUTPUT 722;"OHM"             !2端子抵抗測定を選択します。
50 OUTPUT 722;"NRDGS 5"         !1回のトリガあたり5個の読み取り値を選択します。
60 OUTPUT 722;"TRIG SGL"        !シングル・トリガを生成します。
70 OUTPUT 722;"PAUSE"           !プログラムの実行を中断します。
80 OUTPUT 722;"ACV"             !AC電圧測定を選択します。
90 OUTPUT 722;"NRDGS 10"        !1回のトリガあたり10個の読み取り値を選択します。
100 OUTPUT 722;"TRIG SGL"       !シングル・トリガを生成します。
110 OUTPUT 722;"SUBEND"         !サブプログラムの終わりを示します。
120 END
```

上のサブプログラムを呼び出すと、マルチメータがこのサブプログラムを1行ずつ実行します。行20～60が2端子抵抗の読み取りを5回行い、これらの読み取り値が読み取り値メモリに入れられます。行70に達すると、サブプログラム実行が停止します。その後のCONTコマンドまたはGroup Execute Triggerにより、プログラム実行が再開します。再開すると、行80～100が10回のAC電圧の読み取りを行い、これらの読み取り値が読み取り値メモリに入れられます。サブプログラムが終了すると、合計15個の読み取り値がメモリにあることになります。上のサブプログラムを呼び出すには、次のコマンドを送ります。

```
OUTPUT 722;"CALL OHMAC1"
```

5回の2端子抵抗読み取りが完了したら、AC電圧源をマルチメータに接続してください。CONTコマンドを送るか、(コントローラ)で次のコマンドを実行することで、サブプログラムの実行が再開します。

```
TRIGGER 7
```

PER

周期。マルチメータに対して、入力信号の周期を測定するように命令します。FSOURCEコマンドを使用すれば、入力信号としてAC電圧(デフォルト)、AC+DC電圧、AC電流またはAC+DC電流のどれかを指定できます。

構文 **PER [*max._input*][,*%_resolution*]**

max._input

*max._input*パラメータは、固定レンジ・モードまたはオートレンジ・モードを選択します。レンジは、FSOURCEコマンドにより指定された入力信号の種類に対応します。つまり、ACVが指定された入力信号の場合、*max._input*パラメータがAC電圧測定レンジを指定します。固定レンジを選択するには、ユーザが入力信号の予想ピーク値の絶対値(負の値は使用しない)として*max._input*を指定します。これで、マルチメータが適切なレンジを選択します。各種入力信号に対して使用可能なレンジを示す表は、FUNCコマンドまたはRANGEコマンドを参照してください。

オートレンジ・モードを選択するには、*max._input*に対してAUTOを指定するか、デフォルト・パラメータを指定してください。オートレンジ・モードでは、マルチメータが個々の周期の読み取りの前に入力信号をサンプリングして、適切なレンジを選択します。

電源投入時の*max._input* = 該当せず

デフォルトの*max._input* = AUTO

%_resolution

*%_resolution*は、下に示すように分解能桁数とゲート時間を指定します(*%_resolution*は、読み取り速度にも影響します。詳細については、付録A「仕様」を参照してください)。

%_resolution **パラメータ**	**選択される ゲート時間**	**分解能桁数**
0.00001	1s	7
0.0001	100ms	7
0.001	10ms	6
0.01	1ms	5
0.1	100µs	4

電源投入時の*%_resolution* = 適用なし
デフォルトの*%_resolution* = 0.00001

備考
- 読み取り速度は、入力信号の 1 周期、ゲート時間、またはデフォルトの読み取りタイム・アウトの1.2秒のうち一番長いものです。
- 周波数(および周期)測定は、 入力信号が正または負のスロープで特定の電圧と交差する時点を検出するレベル検出回路を使用して行われます(周波数または周期測定時には、LEVELトリガ・イベント、LEVELサンプル・イベントまたはLINEトリガ・イベントを使用できないのはこのためです)。電源投入時のレベル・トリガ値またはデフォルトのレベル・トリガ値では、正のスロープのゼロ・ボルトが選択されます。レベル・トリガ電圧と結合は、LEVELコマンドを使用して設定することはできません。SLOPEコマンドを使用すれば、正または負のスロープのいずれかを指定できます。
- 最も左の桁は、ほとんどの測定ファンクションで1/2桁ですが、周期測定では0から9までフルの桁が表示されます。
- オートレンジをオンにして行う読み取りは、より長い時間がかかります。これは、周期読み取りの間に(適切なレンジを決めるため)入力信号がサンプリングされるためです。
- 周期(および周波数)測定では、過負荷表示は電圧または電流の振幅が指定された測定レンジに対して大きすぎることを意味します。印加された周期(または周波数)が大きすぎて測定できないということではありません。
- **関連コマンド**: ACBAND、FREQ、FSOURCE、FUNC、RES

例

```
0 OUTPUT 722;"FSOURCE ACI"      !AC電流を入力信号源として選択します。
20 OUTPUT 722;"PER .01"         !周期測定、10mAレンジを選択します。
30 END
```

PRESET

マルチメータを、3つの定義済みステートの1つに構成します。

構文 **PRESET [*type*]**

type

NORM、FASTまたはDIGプリセット・ステートを指定します(これらのパラメータの数値問合せの等価値は、それぞれ1、0および2です)。

電源投入時の*type* = 該当せず
デフォルトの*type* = NORM

NORM

PRESET NORMはRESETと似ていますが、リモート操作用にマルチメータを最適化します。PRESET NORMを実行すると、次のコマンドが実行されます。

```
ACBAND 20,2E+6          MEM OFF(最後のメモリ動作はFIFOに設定されます)
AZERO ON                MFORMAT SREAL
BEEP ON                 MMATH OFF
DCV AUTO                NDIG 6
DELAY -1                NPLC 1
DISP ON                 NRDGS 1,AUTO
FIXEDZ OFF              OCOMP OFF
FSOURCE ACV             OFORMAT ASCII
INBUF OFF               TARM AUTO
LOCK OFF                TIMER 1
MATH OFF                TRIG SYN
```

すべての演算レジスタが0に設定されます。ただし、次を除きます。

```
DEGREE = 20
PERC = 1
REF = 1
RES = 50
SCALE = 1
```

FAST

PRESET FASTは、高速読み取り、メモリへの高速転送、およびメモリからGPIBへの高速転送用に マルチメータを構成します(高速測定の詳細については、第4章の「読み取り速度を上げる」を参照してください)。PRESET FASTを実行すると、次の例外を除いて、PRESET NORMの下に示したコマンドが実行されます。

```
DCV 10
AZERO OFF
DISP OFF
MFORMAT DINT
OFORMAT DINT
TARM SYN
TRIG AUTO
```

DIG

PRESET DIGは、DCVデジタイズ用にマルチメータを構成します(DCVデジタイズは第5章で説明しています)。PRESET DIGを実行すると、次の例外を除いて、PRESET NORMの下に示したコマンドが実行されます。

```
DCV 10
AZERO OFF
DELAY 0
DISP OFF
TARM HOLD
TRIG LEVEL
LEVEL 0,AC
NRDGS 256,TIMER
TIMER 20E-6
APER 3E-6
MFORMAT SINT
OFORMAT SINT
```

備考 • **関連コマンド**: RESET

例
```
OUTPUT 722;"PRESET NORM"      !リモート操作用に構成します。
OUTPUT 722;"PRESET FAST"      !高速読み取り/転送用に構成します。
OUTPUT 722;"PRESET DIG"       !高速DCVデジタイズ用に構成します。
```

PURGE

ステートの削除。保存済みステートを1つメモリから削除します。

構文 **PURGE *name***

name

ステート名。ステート名の最大文字数は10です。名前には、英字、英数字または0～127の範囲の整数が使えます。詳細については、SSTATEコマンドを参照してください。

電源投入時の*name* = なし
デフォルトの*name* = なし、パラメータが必要

備考
- すべての保存済みステートを削除するには、SCRATCHコマンドを使用します。
- **関連コマンド**: DELSUB、SCRATCH

例
```
OUTPUT 722; "PURGE A2"      !保存済みステートA2を削除します。
```

QFORMAT

問合せフォーマット。問合せ応答が(可能な場合常に)数字または英字を含むかどうか、またコマンド・ヘッダが返されるかどうかを指定します。

構文 **QFORMAT [*type*]**

format

*type*パラメータの選択肢は次のとおりです。

type パラメータ	数値 問合せの 等価値	説明
NUM	0	GPIBまたはディスプレイに送られる問合せ応答が、ヘッダなしで(可能な場合)数字のみ。

type パラメータ	数値問合せの等価値	説明
NORM	1	GPIBに送られる問合せ応答がヘッダなしで(可能な場合)数字のみ。ディスプレイに送られる問合せ応答は英字ヘッダと(可能な場合)英字応答を含みます。
ALPHA		GPIBまたはディスプレイのいずれかへ送られる問合せ応答が英字ヘッダと(可能な場合)英字応答を含みます。

電源投入時の*type* = NORM
デフォルトの*type* = NORM

- 本章では英字パラメータに等価の**数値問合せの値**を、個々の該当するコマンドの下に示してあります。DEFKEY?などの一部の問合せコマンドは、指定されたQFORMATに関係なく、英字を返します。同様に、NDIG?などの一部の問合せコマンドは、常に数値応答を返します。

- 問合せコマンドをマルチメータの前面パネルから実行すると、結果はディスプレイにのみ送られます。問合せコマンドをコントローラから実行すると、結果はマルチメータの出力バッファにのみ送られます。問合せ結果は、ASCIIフォーマットで返され、その後出力フォーマットは前に指定されていたタイプ(ASCII、SINTなど)に戻ります。

- **問合せコマンド**　QFORMAT?問合せコマンドは、現在の問合せフォーマットを返します。詳細については、本章の始めにある「問合せコマンド」を参照してください。

- **関連コマンド**: すべての問合せコマンド、OFORMAT

例 NORM

```
10 OUTPUT 722;"QFORMAT NORM"
20 OUTPUT 722;"ARANGE?"
30 ENTER 722;A
40 PRINT A
50 END
```

代表的な応答: `1`

NUM

```
10 OUTPUT 722;"QFORMAT NUM"
20 OUTPUT 722;"ARANGE?"
30 ENTER 722;A
40 PRINT A
50 END
```

代表的な応答: `1`

ALPHA

```
10 OUTPUT 722; "QFORMAT ALPHA"
20 OUTPUT 722; "ARANGE?"
30 ENTER 722;A$
40 PRINT A$
50 END
```

代表的な応答: `ARANGE ON`

R

Rは、RANGEコマンドの省略形です。

構文 **R [*max._input*][,*%_resolution*]**

詳細については、RANGEコマンドを参照してください。

RANGE

RANGEコマンドを使用すると、測定レンジまたはオートレンジ･モードを選択できます。

構文 RANGE [*max._input*][,*%_resolution*]

max._input

*max._input*パラメータは、固定レンジ・モードまたはオートレンジ・モードを選択します。固定レンジを選択するには、ユーザが入力信号の予想ピーク値の絶対値(負の値は使用しない)として*max._input*を指定します。これで、マルチメータが適切なレンジを選択します。オートレンジ･モードを選択するには、*max._input*に対してAUTOを指定するか、デフォルト・パラメータを指定してください。オートレンジ・モードでは、マルチメータが個々の読み取りの前に入力信号をサンプリングして、適切なレンジを選択します。

- 次の表は、*max._input*パラメータと各測定ファンクションに対して選択されるレンジを示します。

DCV:

max._input パラメータ	選択される レンジ	フル・ スケール
–1またはAUTO	オートレンジ	
0～0.12	100mV	120mV
>0.12～1.2	1V	1.2V
>1.2～12	10V	12V
>12～120	100V	120V
>120～1E3	1000V	1050V

ACVまたはACDCV:

max._input パラメータ	選択される レンジ	フル・ スケール
–1またはAUTO	オートレンジ	
0～0.012	10mV	12mV
>0.012～0.12	100mV	120mV
>0.12～1.2	1V	1.2V
>1.2～12	10V	12V
>12～120	100V	120V
>120～1E3	1000V	1050V

OHMまたはOHMF:

max._input パラメータ	選択される レンジ	フル・ スケール
–1またはAUTO	オートレンジ	
0～12	10Ω	12Ω
>12～120	100Ω	120kΩ
>120～1.2E3	1kΩ	1.2kΩ
>1.2E3～1.2E4	10kΩ	12kΩ
>1.2E4 1.2E5	100kΩ	120kΩ
>1.2E5～1.2E6	1MΩ	1.20MΩ
>1.2E6～1.2E7	10MΩ	12MΩ
>1.2E7 1.2E8	100MΩ	120MΩ
>1.2E8 1.2E9	1GΩ	1.2GΩ

DCI:

max._input パラメータ	選択される レンジ	フル・ スケール
–1またはAUTO	オートレンジ	
0～0.12E-6	0.1μA	0.12μA
>0.12E-6～1.2E-6	1μA	1.2μA
>1.2E-6～12E-6	10μA	12μA
>12E-6～120E-6	100μA	120μA
>120E-6～1.2E-3	1mA	1.2mA
>1.2E-3～12E-3	10mA	12mA
>12E-3～120E-3	100mA	120mA
>120E-3～1.2	1A	1.05A

ACIまたはACDCI:

max._input パラメータ	選択される レンジ	フル・ スケール
–1またはAUTO	オートレンジ	
0～0.120E-6	100μA	120μA
>120E-6～1.2E-3	1mA	1.2mA
>1.2E-3～12E-3	10mA	12mA
>12E-3～120E-3	100mA	120mA
>120E-3～1.2	1A	1.05A

DSACまたはDSDC:

		フル・スケール	
max._input パラメータ	選択 される レンジ	SINT フォー マット	DINT フォー マット
0～0.012	10mV	12mV	50mV
>0.012～0.120	100mV	120mV	500mV
>0.120～1.2	1V	1.2V	5.0V
>1.2～12	10V	12V	50V
>12～120	100V	120V	500V
>120～1E3	1000V	1050V	1050V

SSACまたはSSDC:

max._input パラメータ	選択される レンジ	フル・ スケール
0～0.012	10mV	12mV
>0.012～0.120	100mV	120mV
>0.120～1.2	1V	1.2V
>1.2～12	10V	12V
>12～120	100V	120V
>120～1E3	1000V	1050V

電源投入時の***max._input*** = AUTO
デフォルトの***max._input*** = AUTO

%_resolution

デジタイズ・ファンクション(DSAC、DSDC、SSACおよびSSDC)以外のすべてのファンクションで、*%_resolution*パラメータは測定分解能を指定します(*%_resolution*がデジタイ

ズ・コマンドに含まれている場合には、マルチメータは*%_resolution*を無視します)。周波数／周期測定では、ユーザが*%_resolution*を分解能桁数として指定します。他の測定ファンクション(DCV、ACV、ACDCV、OHM、OHMF、DCIおよびACI)では、ユーザが*%_resolution*を*max._input*パラメータのパーセントとして指定します。これで、マルチメータが*%_resolution*に*max._input*を掛けて測定の分解能を求めます。

例えば、最大予想入力が10Vで、1mVの分解能が必要と仮定します。*%_resolution*を求めるには、次の式を使用します。

%_resolution = (実際の分解能/最大入力)×100

この例では、式は次のようになります。

%_resolution = (0.001/10)×100 = 0.0001×100 = 0.01

注記 オートレンジを使用する場合、マルチメータが%_resolutionパラメータに、選択されたレンジのフル・スケール読み取り値を掛けます。その結果が、最小分解能です。少なくとも最小分解能になり、多くの場合、追加の分解能桁数が加わります。

電源投入時の*%_resolution* = なし　電源投入時、分解能はNPLCコマンドによって決まり、この場合8 1/2桁になります(NDIGの電源投入時の値はディスプレイの1桁をマスクするため、マルチメータは7 1/2桁しか表示しません。NDlG 8コマンドを使用すれば、8 1/2桁をすべて表示できます。詳細は、NDIGコマンドを参照してください)。

デフォルトの *%_resolution*:

周波数または周期測定では、デフォルトの*%_resolution*は0.00001で、これにより1sのゲート時間と7桁の分解能が選択されます。

サンプリングしたACVまたはACDCVでは、デフォルトの*%_resolution*は、SETACV SYNCでは0.01%、SETACV RNDMでは0.4%です。

他のすべての測定ファンクションでは、デフォルトの分解能は現在の積分時間で決まります。

備考

- **問合せコマンド**　RANGE?問合せコマンドは現在の測定レンジを返します(RANGE?では、オートレンジ・モードは分かりません。オートレンジ・モードを判別するには、ARANGE?コマンドを使用してください)。詳細については、本章の始めにある「問合せコマンド」を参照してください。
- **関連コマンド**: ARANGE、FUNC、R

例 次のプログラムでは、行10によって行30の*%_resolution*による分解能の設定が可能になっています。行30により指定されている分解能は10mΩです。

```
10 OUTPUT 722;"NPLC 0"              !PLC数を最大に設定します。
20 OUTPUT 722;"OHM"                 !2端子抵抗を選択します。
30 OUTPUT 722;"RANGE 800,.00125"    !800Ω MAX、100mΩを選択します。
40 END                              !分解能
```

RATIO

RATIOコマンドは、マルチメータに対して、Ω **Sense**端子に印加されるDC基準電圧と、**Input**端子に印加される信号電圧を測定するように命令します。その後、マルチメータが次のように比を計算します。

$$比 = \frac{信号電圧}{DC\ 基準電圧}$$

構文 **RATIO [*control*]**

control

control パラメータ	数値問合せの等価値	説明
OFF	0	比測定をオフにします。
ON	1	現在の測定ファンクション(DCV、ACVまたはACDCV)を使用した比測定をオンにします。

電源投入時の*control* = OFF
デフォルトの*control* = ON

備考

- Ω **Sense LO**端子と**Input LO**端子には共通の基準が必要で、電圧差が0.25Vが超えてはいけません。
- 信号電圧は、DCV、ACV、またはACDCV測定ファンクションを使用して測定できます(ACVまたはACDCVの場合、3つの測定法、ANA、RNDMまたはSYNCのいずれかを使用できます)。マルチメータは常に、DCVを基準電圧測定に使用します。測定可能な基準電圧範囲は、±12VDC(オートレンジのみ)です。比測定を指定するには、ユーザが最初に測定ファンクション(ACVまたはACDCVの測定法)を設定し、次にRATIOコマンドを使用して比測定をオンにします(下の例を参照)。
- **問合せコマンド** RATIO?問合せコマンは、現在の比モードを返します。詳細については、本章の始めにある「問合せコマンド」を参照してください。
- **関連コマンド**: ACDCV、ACV、DCV、SETACV

例

```
10 OUTPUT 722;"PRESET NORM"    !読み取りを中断します。NRDGS=1
20 OUTPUT 722;"ACV"            !AC電圧測定を選択します。
30 OUTPUT 722;"SETACV SYNC"    !同期ACV測定
40 OUTPUT 722;"RATIO ON"       !比測定をオンにします。
50 OUTPUT 722;"TRIG SGL"       !測定をトリガします。
60 ENTER 722;A                 !比を入力します。
70 PRINT A                     !比を出力します。
80 END
```

RES

分解能。読み取り分解能を指定します。

構文 **RES [*%_resolution*]**

%_resolution

周波数／周期測定では、*%_resolution*パラメータは下に示すように分解能桁数とゲート時間を指定します(*%_resolution*は読み取り速度にも影響します。詳細については、付録A「仕様」を参照してください)。周波数または周期測定で*%_resolution*パラメータを省略すると、マルチメータは0.00001を使用します。

%_resolution **パラメータ**	**選択される ゲート時間**	**分解能桁数**
0.00001	1s	7
0.0001	100ms	7
0.001	10ms	6
0.01	1ms	5
0.1	100µs	4

サンプリングしたACVまたはACDCVの場合は、ランダム・サンプリング(SETACV RNDM)の分解能は4.5桁に固定され、変更できません。同期サンプリング(SETACV SYNC)の場合は、*%_resolution*パラメータは、0.001 = 7.5桁、0.01 = 6.5桁、0.1 = 5.5桁、1= 4.5桁となります。

他のすべてのファンクションの場合(DSAC、DSDC、SSACおよびSSDCを除く。これらのファンクションでは*%_resolution*は無視されます)、マルチメータが*%_resolution*に現在の測定レンジ(1V、10V、100Vなど)を掛けて分解能を求めます。*%_resolution*パラメータを計算するには、次の式を使用します。

%_resolution = (実際の分解能/レンジ)×100

例えば、10VレンジでDC電圧を測定していて、100µVの分解能が必要だと仮定します。式は次のようになります。

%_resolution = (0.0001/10)×100 = 0.001

電源投入時の*%_resolution*なし　電源投入時、分解能はNPLCコマンドによって決まり、この場合8 1/2桁になります(NDIGの電源投入時の値はディスプレイの1桁をマスクするため、マルチメータは7 1/2桁しか表示しません。NDlG 8コマンドを使用すれば、8 1/2桁をすべて表示できます)。

デフォルトの *%_resolution*:

周波数または周期測定では、デフォルトの*%_resolution*は0.00001で、これにより1sのゲート時間と7桁の分解能が選択されます。

サンプリングしたACVまたはACDCVでは、デフォルトの*%_resolution*は、SETACV SYNCでは0.01%、SETACV RNDMでは0.4%です。

他のすべての測定ファンクションでは、デフォルトの分解能は現在の積分時間で決まります。

備考

- アナログ測定では、RESコマンドの*%_resolution*パラメータの動作は、ファンクション・コマンド(FUNC、ACV、DCVなど)またはRANGEコマンドの*%_resolution*とはやや異なります。RESコマンドと合わせて使用すると、*%_resolution*にレンジが掛けられて実際の分解能が求められます。ファンクション・コマンドまたはRANGEコマンドと合わせて使用すると、*%_resolution*にそのコマンドの*max._input*パラメータが掛けられます。*max._input*パラメータは、測定レンジの値である場合と、そうでない場合があります。
- **問合せコマンド**　RES?問合せコマンドは指定されている*%_resolution*を返します。詳細については、本章の始めにある「問合せコマンド」を参照してください。
- **関連コマンド**: ACDCI、ACDCV、ACI、ACV、APER、DCI、DCV、FREQ、FUNC、NPLC、OHM、OHMF、PER、RANGE

例　次のプログラムでは、行10によって行30の*%_resolution*により分解能の設定が可能になっています。

```
0 OUTPUT 722;"NPLC 0"           !PLC数を最小に設定します。
20 OUTPUT 722;"DCV 6,"          !DC電圧、10Vレンジを選択します。
30 OUTPUT 722;"RES .001"        !10Vレンジで100µVの分解能
40 END
```

次のプログラムでは、行10がPLC数を1000に設定します。これは最大分解能(7.5桁)に対応し、行30のRESコマンドが測定に影響することを防ぎます。行30が要求する分解能は10mΩです。ただし、行10のために、実際の分解能は100µΩになります。

```
10 OUTPUT 722;"NPLC 1000"       !PLC数を最大に設定します。
20 OUTPUT 722;"OHM 1E3          !2端子抵抗、1kΩレンジを選択します。
30 OUTPUT 722;"RES .001         !10mΩ分解能を要求します。
40 END
```

RESET

電源を入れなおさなくてもマルチメータを電源投入時のステートに設定できます。

構文　**RESET**

備考

- RESETコマンドは以下を行います。

読み取りプロセスを中断します。
エラー・レジスタと補助エラー・レジスタをクリアします。
電源投入時のSRQビット(ビット3)を除いてステータス・レジスタをクリアします。
読み取り値メモリをクリアします。

さらに、RESETコマンドは次のコマンドも実行します。

ACBAND 20,2E6	MFORMAT SREAL
AZERO ON	MMATH OFF
DCV AUTO	NDIG 7
DEFEAT OFF	NPLC 10
DELAY -1	NRDGS 1,AUTO
DISP ON	OCOMP OFF
EMASK 32767 (すべてオン)	OFORMAT ASCII QFORMAT NORM
END OFF	RATIO OFF
EXTOUT ICOMP,NEG	RQS 0
FIXEDZ OFF	SETACV ANA
FSOURCE ACV	SLOPE POS
INBUF OFF	SSRC LEVEL,AUTO
LEVEL 0,AC	SWEEP 100E-9,1024
LFILTER OFF	TARM AUTO
LFREQ (電源ライン周波数は50または60Hzに丸める)	TBUFF OFF
LOCK OFF	TIMER 1
MATH OFF	TRIG AUTO
MEM OFF (最後のメモリ動作はFIFOに設定)	

次を除いて、演算レジスタはすべて0に設定されます。

```
DEGREE = 20       REF = 1
SCALE  = 1        RES = 50
PERC   = 1
```

- RESETはリモートからも使用できますが、本来前面パネルから使用することを意図しています。RESETは、ローカル操作を始めるために適した状態にマルチメータを構成します。RESETコマンドをアルファベット・メニューから実行すると、マルチメータが上に示すようにリセットされます。ただし、シフト+前面パネル**Reset**キーを押しても、マルチメータの電源を入れ直したのと同じことになります。この場合は、現在のステートがステート0として保存され、圧縮されたサブプログラムがあれば破棄され、保存済み読み取り値も破棄され、電源投入時のSRQビットがステータス・レジスタ内で設定されて、電源投入時のシーケンスが実行されます。
- RESETコマンドをリモートから送ろうとすると、マルチメータがビジーまたはGPIBバスがホールドされていることがあり得ます。どちらの場合でも、マルチメータはリモートRESETコマンドに直ちには応答しません。このため、GPIBデバイス・クリア・コマンドを送った後に、マルチメータのRESETコマンドを送る必要があります。これを、下の例に示します。
- 関連コマンド: PRESET

例

```
10 CLEAR 722                !マルチメータを直ちにクリアします。
20 OUTPUT 722;"RESET"       !マルチメータをリセットします。
30 END
```

REV?

リビジョンの問合せ。カンマで区切られた2つの数値を返します。１番目の数値は、マルチメータのマスタ・プロセッサのファームウェア・リビジョンです。2番目の数値は、スレーブ・プロセッサのファームウェア・リビジョンです。

構文 **REV?**

例
```
10 OUTPUT 722; "REV?"        !ファームウェア・リビジョン番号を読み取ります
20 ENTER 722; A, B           !数値を入力します。
30 PRINT A, B                !数値を出力します。
40 END
```

RMATH

演算の呼び出し。演算レジスタの内容を読み取って、返します。

構文 **RMATH [*register*]**

register

*register*パラメータの選択肢は次のとおりです。

register パラメータ	数値問合せの 等価値	レジスタの内容
DEGREE	1	FILTERとRMSの時定数
LOWER	2	STATS内の最小読み取り値
MAX	3	PFAIL演算の上限値
MEAN	4	STATSの読み取り値の平均
MIN	5	PFAIL演算の下限値
NSAMP	6	STATS内のサンプル数
OFFSET	7	NULLおよびSCALE演算の減数
PERC	8	PERC演算の%値
REF	9	DB演算の基準値
RES	10	DBM演算の基準インピーダンス
SCALE	11	SCALE演算の除数
SDEV	12	STATSの標準偏差
UPPER	13	STATSの最大読み取り値
HIRES	14	どの演算にも使用されず(エクストラ・レジスタ)
PFAILNUM	15	障害が生じる前にPFAILをパスした読み取り値の個数

電源投入時の*register* = なし
デフォルトの*register* = DEGREE

備考

- 演算レジスタの内容は、指定された出力フォーマットに関係なくASCII出力フォーマットで出力されます。その後、出力フォーマットは前に指定されていたフォーマット(SINT、DINT、SREAL、DREALまたはASCII)に戻ります。
- **関連コマンド**: MATH、MMATH、SMATH

例

```
10 OUTPUT 722;"TRIG HOLD"      !トリガを中断します。
20 OUTPUT 722;"MEM FIFO"       !読み取り値メモリをFIFOモードでオンにします。
30 OUTPUT 722;"NRDGS 10"       !1回のトリガあたり10個の読み取り値
40 OUTPUT 722;"DCV 3"          !DC電圧、10Vレンジ
50 OUTPUT 722;"MATH STAT"      !統計演算をオンにします。
60 OUTPUT 722;"TRIG SGL"       !マルチメータを1回トリガします。
70 OUTPUT 722;"RMATH SDEV"     !標準偏差を読み取ります。
80 ENTER 722;A                 !標準偏差を入力します。
90 PRINT A                     !標準偏差を出力します。
100 END
```

RMEM

メモリの呼び出し。読み取り値メモリに保存されている読み取り値または読み取り値グループの値を読み取って返します。RMEMでは、保存されている読み取り値は元のままです(メモリからクリアされません)。

構文 **RMEM [*first*][,*count*] [,*record*]**

first

開始読み取り値を指定します。

電源投入時の*first* = なし
デフォルトの*first* = 1

count

*first*で始まる、呼び出す読み取り値の個数を指定します。

電源投入時の*count* = なし
デフォルトの*count* = 1

record

読み取り値を呼び出すレコードを指定します。レコードは、NRDGSコマンドによって指定された読み取り値の個数に対応します。例えば、NRDGSが1回のトリガあたり3個の読み取り値を指定していれば、個々のレコードには3個の読み取り値が含まれます。

電源投入時の*record* = なし

デフォルトの *record* = 1

備考

- RMEMコマンドは、読み取り値メモリを自動的にオフにします(MEM OFF)。これは、保存済みの読み取り値はすべてそのままで、新しい読み取り値は保存されないということです。MEM CONTコマンドを使用すれば、保存済み読み取り値を破棄しないで、読み取り値メモリを再度オンにできます。
- マルチメータは、読み取り値メモリ内の個々の読み取り値に番号を割り当てます。最も新しい読み取り値には、最も低い番号(1)が割り当てられ、最も古い読み取り値はは最も高い番号になります。FIFOモードまたはLIFOモードのどちらを使用しているかに関係なく、番号はこの方法で割り当てられます。レコードにもこの方法で番号が付けられます。最も新しいレコードがレコード番号1になります。
- RMEM コマンドを前面パネルから実行すると、一度に 1 個ずつ読み取り値がディスプレイへコピーされます。最初の読み取り値を表示した後、上または下矢印キーを使用して他の読み取り値を表示できます。読み取り値番号(ディスプレイの左側)と読み取り値(ディスプレイの右側)を表示するには、左または右矢印キーを使用します。
- RMEMコマンドの他に、"暗黙の読み取り"を使用して読み取り値を呼び出すこともできます。詳細については、第4章の「読み取り値のリコール」を参照してください。
- **関連コマンド**: MCOUNT?、MEM、MFORMAT、MSIZE、NRDGS

例

```
10 OUTPUT 722;"TARM HOLD"       !トリガを中断します。
20 OUTPUT 722;"DCV"             !DC電圧測定
30 OUTPUT 722)"TRIG AUTO"       !自動トリガ
40 OUTPUT 722;"NRDGS 3 ,AUTO"   !1つのサンプル・イベントあたり3個の読み取り値(AUTO)
50 OUTPUT 722;"MEM FIFO"        !読み取り値メモリをFIFOモードでオンにします。
60 OUTPUT 722;"TARM SGL, 10"    !読み取り値のグループが10個
70 OUTPUT 722;"RMEM 1,3,6"      !6番目のグループの1番目～3番目の読み取り値を読み取ります。
80 ENTER 722;A,B,C              !読み取り値をA、BおよびC変数に入力します。
90 PRINT A,B,C                  !読み取り値を出力します。
100 END
```

RQS

サービス要求。1つまたは複数のステータス・レジスタ条件をオンにします。条件がオンされていて、この条件が発生すると、GPIB SRQ行が真に設定されます。

構文 **RQS [*value*]**

value

条件をオンにするには、その10進重みを値パラメータとして指定します。複数の条件の場合は、重みの和を指定します。条件とその重みは次のとおりです。

10進重み	ビット番号	オンされる条件
1	0	プログラム・メモリ実行完了
2	1	上限または下限超過
4	2	SRQコマンド実行
8	3	電源投入時のSRQ
16	4	命令準備完了
32	5	エラー (エラー・レジスタを確認)
64	6	サービス要求(このビットはオフにできません)
128	7	データ使用可能

電源投入時の*value*: 電源投入時のSRQがオンにされている場合、電源が切断されるとvalue = 8。それ以外はvalue = 0。
デフォルトの*value* = 0 (どの条件もオンでない)

備考

- ビット5を設定するエラーは、EMASKコマンドを使用して制御できます。
- 電源投入時のSRQビットは、不揮発性メモリに保存されます。他のビットはすべて、電源投入時にクリアされます。
- **問合せコマンド**　RQS?問合せコマンドは、ステータス・レジスタ内のオンにされたすべてのビットの重み和を返します。
- **関連コマンド**: CSB、SPOLL(GPIBコマンド)、STB?

例

```
OUTPUT 722;"RPS 4"       !前面パネルSRQ条件をオンします。
OUTPUT 722;"RQS 40"      !電源投入時のSRQ(8)条件とERROR(32)条件をオンにします。
OUTPUT 722;"RQS 255"     !すべての条件をオンにします。
OUTPUT 722;"RQS 0"       !すべての条件をオフにします。
```

RSTATE

ステートの呼び出し。保存済みステートをメモリから呼び出して、マルチメータをこのステートに構成します。ステートは、SSTATEコマンドを使用して保存します。

構文 **RSTATE [*name*]**

name

ステート名。ステート名の最大文字数は10です。名前には、英字、英数字、または0～127の範囲の整数が使用できます。詳細については、SSTATEコマンドを参照してください。

電源投入時の*name* = なし
デフォルトの*name* = 0

備考
- マルチメータの電源が切断されると、常に現在の状態がステート0に保存されます。停電の後、RSTATE 0を実行することでマルチメータを前のステートに構成できます。
- 保存済みステートでNULLリアルタイム演算がオンの場合、ステートを呼び出すと、最初の読み取り値がOFFSETレジスタに入れられます(詳細については、第4章の「ヌル」を参照してください。
- 前面パネルから、**Recall State**キーを押して、上および下矢印キーを使用してすべての保存済みステートの名前を表示することができます。必要なステートが見つかったら、**Enter**キーを押してそのステートを呼び出します。
- **関連コマンド**: MSIZE、PURGE、SCRATCH、SSTATE

例
```
OUTPUT 722; "RSTATE B2"      !B2という名前の保存済みステートを呼び出します。
```

SCAL

これは校正コマンドです。詳細については、『3458A校正マニュアル』を参照してください。

SCRATCH

サブプログラムと保存済みステートのすべてをメモリからクリアします。

構文 **SCRATCH**

備考
- 個々のサブプログラムは、DELSUBコマンドを使用してクリアできます。個々のステートは、PURGEコマンドを使用してクリアできます。
- **関連コマンド**: DELSUB、PURGE、RSTATE、SSTATE、SUB

例
```
OUTPUT 722;"SCRATCH"        !すべてのサブプログラムと保存済みステートをクリアします。
```

SECURE

セキュリティ・コード。校正責任者がセキュリティ・コードを入力すれば、偶然の、あるいは許可されていない校正または自動校正を防ぐことができます(自動校正の詳細については、ACALコマンドを参照してください)。

構文 **SECURE *old_code, new_code*[,*acal_secure*]**

old_code

これは本マルチメータの前のセキュリティ・コードです。本マルチメータは、セキュリ

ティ・コードを4358に設定して出荷されています。

new_code

これは新しいセキュリティ・コードです。このコードは–2.1E9～2.1E9の整数です。指定した数値が整数でない場合は、マルチメータが整数値に丸めます。

acal_secure

自動校正にセキュリティをかけることができます。次の選択肢があります。

acal_secure パラメータ	数値問合せの等価値	説明
OFF	0	自動校正セキュリティをオフにします。自動校正にコードは不要です。
ON	1	自動校正セキュリティをオンにします。自動校正を実行するにはセキュリティ・コードが必要です(例えば、ACALを参照)。

電源投入時の*acal_secure* = 前に指定された値(ONが工場設定です)
デフォルトの*acal_secure* = OFF

備考

- *new_code*に対して0を設定するとセキュリティ機能がオフになり、校正または自動校正を実行するためにセキュリティ・コードを入力する必要がなくなります。
- 前面パネルの**Last Entry**キーでは、前に実行されたSECUREコマンドで使用されたコードは表示されません。
- **関連コマンド**: ACAL、CAL、CALNUM?、CALSTR、SCAL

例 コードの変更

```
OUTPUT 722;"SECURE 3458,4448,ON"  !工場セキュリティ・コードを4448に変更し、
!自動校正セキュリティをオンにします。
```

セキュリティをオフにする

```
OUTPUT 722;"SECURE 3458,0"     !校正および自動校正のセキュリティをオフにします。
```

SETACV

ACVの設定。ACまたはAC+DC電圧測定に使用するRMS変換法を選択します。

構文 **SETACV [*type*]**

type

*type*パラメータは、測定法を選択するために使用します。測定法は、アナログ、ランダ

ム・サンプリング または同期サンプリングです。選択肢は次のとおりです。

type パラメータ	数値問合せの 等価値	説明
ANA	1	アナログRMS変換
RNDM	2	ランダム・サンプリング変換
SYNC	3	同期サンプリング変換

電源投入時の***type*** = ANA
デフォルトの***type*** = ANA

備考
- 帯域幅の制限は、選択する変換法により異なります。詳細については、付録A「仕様」を参照してください。
- **問合せコマンド**　SETACV?問合せコマンドは、現在のAC測定法を返します。詳細については、本章の始めにある「問合せコマンド」を参照してください。
- **関連コマンド**: ACBAND、ACDCV、ACV、FUNC、SSRC

例
```
10 OUTPUT 722; "SETACV SYNC"    !同期サンプリング(DC結合)を指定します。
20 OUTPUT 722;"ACDCV"           !AC+DC電圧測定を選択します。
30 END
```

SLOPE

SLOPEはLEVELコマンドとともに使用し、レベル検出回路が使用する、信号のスロープを指定します。

構文 **SLOPE [*slope*]**

slope

レベル検出回路が使用する、入力信号の立ち上がりまたは立ち下がりスロープを選択します。次の選択肢があります。

slope パラメータ	数値問合せの 等価値	説明
NEG	0	立ち下がりスロープを選択します。
POS	1	立ち上がりスロープを選択します。

電源投入時の***slope*** = POS
デフォルトの***slope*** = POS

備考
- **問合せコマンド**　SLOPE?問合せコマンドは、現在のスロープを返します。詳細については、本章の始めにある「問合せコマンド」を参照してください。
- **関連コマンド**: LEVEL、LFILTER、NRDGS、SSRC、TRIG

例 `OUTPUT 722;"SLOPE POS"`　　!レベル検出用に立ち上がりスロープを
!選択します。

SMATH

演算の保存。数値を演算レジスタに入れます。

構文 **SMATH [*register*][,*number*]**

register

書き込みのできるレジスタは次のとおりです。

register パラメータ	数値問合せの等価値	レジスタの内容	電源投入時の値
DEGREE	1	FILTERおよびRMSの時定数	20
LOWER	2	STATS内の最小読み取り値	0
MAX	3	PFAIL演算の上限値	0
MEAN	4	STATSの読み取り値の平均	0
MIN	5	PFAILの下限値	0
NSAMP	6	STATS内のサンプル数	0
OFFSET	7	NULLおよびSCALE演算の減数	0
PERC	8	PERC演算の%値	1
REF	9	DB演算の基準値	1
RES	10	DBM演算の基準インピーダンス	50
SCALE	11	SCALE演算の除数	1
UPPER	13	STATSの最大読み取り値	0
HIRES	14	どの演算にも使用されず	0
PFAILNUM	15	障害が現れる前にPFAILをパスした読み取り値の個数	0

デフォルトの***register*** = DEGREE
電源投入時の***register*** = 上の一覧を参照

number

*number*パラメータは、レジスタに入れられる値です。

デフォルトの*number* = 最後の読み取り値
電源投入時の*number* = 上の一覧を参照

備考
- SMATHコマンドを使用すれば、読み取り値(UPPER、LOWERなど)を保存するレジスタの1つに数値を入れることができます。ただし、この値は、対応する演算機能がオンになっている場合(STATSなど)、読み取り値で置換されます。
- –1(マイナス1)を使用して*number*をデフォルト設定することはできません。–1を指定すると、実際に–1をレジスタに書き込むことになります。
- **関連コマンド**: MATH、MMATH、RMATH

例

```
OUTPUT 722;"SMATH 11,1E-3" !"1E-3"をスケール・レジスタに入れます。
```

次のプログラムでは、行10と行20は、抵抗測定のための構成を行います。行30が、抵抗測定をトリガします。行40は、*number*パラメータをデフォルトに設定し、抵抗読み取り値をRESレジスタに保存します。行50は、ユーザに対して電圧源をマルチメータに接続するように指示します。行80はDBM演算をオンにします。このプログラムは、抵抗に供給される電力をdB単位で(DBM演算の結果)表示します。

```
10 OUTPUT 722;"PRESET NORM"        !TARM AUTO、TRIG SYN、NRDGS 1、AUTO
20 OUTPUT 722;"OHM"                !2端子抵抗を選択します。
30 OUTPUT 722;"TRIG SGL"           !1回トリガします。
40 OUTPUT 722;"SMATH RES"          !読み取り値をRESレジスタに入れます。
50 DISP "CONNECT SOURCE; PRESS CONT"!ユーザ・プロンプト
60 PAUSE                           !プログラム実行を中断します。
70 OUTPUT 722;"ACV"                !AC電圧を選択します。
80 OUTPUT 722;"MATH DBM"           !DBM演算をオンにします。
90 OUTPUT 722;"TRIG AUTO"          !自動的にトリガします。
100 END
```

SRQ

サービス要求。マルチメータのステータス・レジスタのビット2を設定します。SRQをアサートする(RQS 4コマンド)ためにビット2がオンになっている場合、SRQコマンドを実行するとGPIB SRQ行が設定されます。

構文 **SRQ**

- **関連コマンド**: CSB、EXTOUT、RQS、SPOLL(GPIBコマンド)、STB?

例

```
10 OUTPUT 722;"RQS 4"        !SRQをアサートするためにスタータス・レジスタのビット2を
                             !オンにします。
20 OUTPUT 722;"SRQ"          !ビット2を設定し、SRQをアサートします。
30 END
```

SSAC、SSDC

サブサンプリング。サブサンプリングした電圧測定(デジタイズ)のためにマルチメータを構成します。SSACファンクションでは、入力波形のAC成分のみが測定されます。SSDCファンクションでは、波形のAC成分とDC成分の合成が測定されます。それ以外は、2つのファンクションは同じです。サブサンプリングした測定を行うためには、入力信号が周期的(繰り返し波形)でなければなりません。サブサンプリングした測定では、トラック/ホールド回路(2nsのアパーチャ)と、広帯域幅入力パス(12MHz帯域幅)が使用されます。

構文 **SSAC [*max._input*] [,*% resolution*]**

SSDC [***max._input***] **[,*%_resolution*]**

max._input

測定レンジを選択します(サブサンプリングした測定ではオートレンジは使用できません)。レンジを選択するには、*max._input*を入力信号の予想ピーク振幅として指定します。こうすると、マルチメータが正しいレンジを選択します。次の表は、*max._input*パラメータと選択されるレンジを示します。

max._input パラメータ	選択されるレンジ	フル・スケール
0～0.012	10mV	12mv
>0.012～0.120	100mV	120mV
>0.120～1.2	1V	1.2V
>1.2～12	10V	12V
>12～120	100V	120V
>120～1E3	1000V	1050V

電源投入時の*max._ input* = 該当せず
デフォルトの*max._input* = 10V

% resolution

SSACコマンドまたはSSDCコマンドとともに使用した場合、マルチメータはこのパラメータを無視します。このパラメータは、他のファンクション・コマンド(FUNC、ACI、DCVなど)とコマンド構文の一貫性を保つために使用が許されています。

備考

- オートゼロおよびオートレンジは、サブサンプリングした測定には機能しません。SSACコマンドまたはSSDCコマンドを実行すると、オートゼロとオートレンジの動作は中断されます。
- 直接サンプリングの場合と同様に、最大でレンジの500%までのレベル・トリガ電圧を指定できます。ただし、SINTフォーマットでは、レンジの120%を超えるサンプルは処理できません。
- SSAC または SSDC コマンを実行したときに読み取り値メモリがオフの場合、マルチ

メータが出力フォーマットをSINTに自動的に設定します(メモリ・フォーマットは変更されません)。後で、別の測定ファンクションに変更すると、出力フォーマットは前に指定されていたフォーマットに戻ります。サブサンプリングして、サンプルをGPIBに直接出力する場合は、SINT出力フォーマットを使用する必要があります。ただし、サンプルが最初に読み取り値メモリに入れられる場合には、どの出力フォーマットでも使用できます(次の備考を参照)。これを行うには、SSACコマンドまたはSSDCコマンドを実行する前に、読み取り値メモリをオンにしておく必要があります(読み取り値メモリがオンのときにSSACまたはSSDCを実行しても、出力フォーマットはSINTに変更しません)。

- 読み取り値メモリがオンの状態でサブサンプリングを行う場合、読み取り値メモリがFIFOモードで、空であり(MEM FIFOを実行すると読み取り値メモリはクリアされます)、トリガ・アーム・イベントが発生する前にメモリ・フォーマットがSINTになっていなければなりません。そうでない場合、トリガ・アーム・イベントが発生するとマルチメータがSETTINGS CONFLICTエラーを生成し、サンプルは取得されません。

- サブサンプリングの場合、トリガ・イベントとサンプル・イベントは無視されます(これらのイベントは、第4章で説明しています)。サブサンプリングに適用されるトリガ・イベントは、トリガ・アーム・イベント(TARM コマンド)と同期信号源イベントだけです。

- サブサンプリングでは、入力波形の2周期以上でサンプルが取得されます。サンプルが読み取り値メモリ(MEMコマンド)へ直接送られる場合には、マルチメータがサンプルを自動的に再構成して、合成波形を生成します。サンプルが出力バッファに送られる場合は、コントローラがアルゴリズムを使用して合成波形を再構成する必要があります。このアルゴリズム用のパラメータは、SSPARM?コマンドにより確認できます。

- サンプル間の*effective_interval*と取得される合計サンプル数は、SWEEPコマンドにより指定します(サブサンプリングの場合にはNRDGSコマンドは使用できません)。サブサンプリングでは、指定した*effective_interval*を得るために必要な数量の入力信号の周期をマルチメータが使用します。サブサンプリングの最小*effective_interval*は10nsです(このプロセスの詳細については、第5章の「サブ・サンプリング」を参照してください)。

- **関連コマンド**: DSAC、DSDC、FUNC、ISCALE?、LEVEL、LFILTER、MEM FIFO、SLOPE、PRESET FAST、PRESET DIG、SSDC、SSPARM?、SSRC、SWEEP、TARM

例 次のプログラムでは、サブサンプリングされたデータが、必要なSINTメモリ・フォーマットを使用して読み取り値メモリに送られます。マルチメータが、これらのサンプルを正しい順序でメモリに入れます。次にサンプルは、DREA L 出力フォーマットを使用してコントローラへ転送されます(最初にサブサンプリングされたデータを読み取り値メモリに入れる場合、使用するフォーマットはSINT出力フォーマットに限定されません)。

```
10 OPTION BASE 1                   !番号付けは1から始まります。
20 REAL Samp(1:200) BUFFER         !バッファ配列を作成します。
30 ASSIGN @Dvm TO 722              !マルチメータ・アドレスを割り当てます。
40 ASSIGN @Samp TO BUFFER Samp(*)  !バッファを割り当てます。
50 OUTPUT @Dvm;"PRESET FAST"       !TARM SYN、TRIG AUTO、DINTフォーマット
60 OUTPUT @Dvm; "MEM FIF0"         !FIFO読み取り値メモリ
70 OUTPUT @Dvm; "MFORMAT SINT"     !SINTメモリ・フォーマット
80 OUTPUT @Dvm; "OFORMAT DREAL"    !倍精度実数出力フォーマット
```

```
90 OUTPUT @Dvm; "SSDC 10"          !サブサンプリング、10Vレンジ、DC結合
100 OUTPUT @Dvm; "SWEEP 5E - 6,200"!5µsの実効、間隔、200個のサンプル
110 TRANSFER @Dvm TO @Samp;WAIT    !サンプルをコントローラ・バッファへ転送します。
120 FOR I=1 TO 200
130 IF ABS(Samp(I))=1E+38 THEN     !過負荷を検出します。
140 PRINT "Overload Occurred"      !過負荷メッセージを出力します。
150 ELSE                           !過負荷がない場合:
160 Samp(I)=DROUND(Samp(I),5)      !5桁に丸めます。
170 PRINT Samp(I)                  !個々のサンプルを出力します。
180 END IF
190 NEXT I
200 END
```

以下のページのプログラムでは、SSACコマンドを使用してピーク値5Vの10kHz信号をデジタイズしています。SWEEPコマンドは、マルチメータに対して、2µsの*effective_interval* (Eff_int変数)で1000個のサンプルを取得するように命令します(Num_samples変数)。測定では、同期信号源イベント(入力信号からのトリガ、0%、AC結合、正のスロープ)に対してデフォルトのレベル・トリガが使用されます。行120は、SYNイベントを生成し、サンプルをコンピュータに直接転送します。行240～行410は、サブサンプリングしたデータをソートして合成波形を生成します。合成波形は、Wave_form配列に保存されます。

```
10 OPTION BASE 1                   !配列の番号付けは1から始まります。
20 INTEGER Num_samples,Inc,I,J,K,L!変数を宣言します。
30 Num_samples=l000                !サンプル数を指定します。
40 Eff_int=2.0E-6                  !実効間隔を指定します。
50 INTEGER Int_samp(1:1OOO) BUFFER!整数バッファを作成します。
60 ALLOCATE REAL Wave_form(1:Num_samples)!ソートされたデータの配列を作成します。
70 ALLOCATE REAL Samp(1:Num_samples)!サンプル用の配列を作成します。
80 ASSIGN @Dvm TO 722              !マルチメータ・アドレスを割り当てます。
90 ASSIGN @Int_samp TO BUFFER Int_samp(*)!バッファ I/Oパス名を割り当てます。
100 OUTPUT @Dvm;"PRESET FAST;LEVEL;SLOPE;SSRC LEVEL;SSDC 10"
101 !高速演算、TARM SYN、レベルSYNC信号源0V、正のスロープ
105 !(デフォルト値)サブサンプリング(SINT出力フォーマット)、10Vレンジ
110 OUTPUT @Dvm;"SWEEP ";Eff_int,Num_samples
115 !2µsの実効間隔、1000個のサンプル
120 TRANSFER @Dvm TO @Int_samp;WAIT!SYNイベント、読み取り値を整数配列に
121 !転送します。コンピュータの整数フォーマットはSINTと同じため、
125 !ここでは、データ変換は不要です(整数配列は必要です)。
130 OUTPUT @Dvm;"ISCALE?"          !SINTフォーマットのスケール・ファクタを問い合わせます。
140 ENTER @Dvm; S                  !スケール・ファクタを入力します。
150 OUTPUT @Dvm;"SSPARM?"          !サンサンプリング・パラメータを問い合わせます。
160 ENTER @Dvm;N1,N2,N3            !サンサンプリング・パラメータを入力します。
170 FOR I=1 TO Num_samples
180 Samp(I)=Int_samp(I)            !個々の整数読み取り値を実数に変換します。
190 !フォーマット(次の行での整数オーバフローを防ぐために必要)
190 R=ABS(Samp(I))                 !絶対値を使用してOVLDがないか確認します。
200 IF R>=32767 THEN PRINT "OVLD" !OVLDの場合、過負荷メッセージを出力します。
210 Samp(I)=Samp(I)*S              !読み取り値にスケール・ファクタを掛けます。
```

```
220 Samp(I)=DROUND(Samp(I),4)     !4桁に丸めます。
230 NEXT I
235 !--------------------------SORT SAMPLES------------------------------
240 Inc=N1+N2                     !バーストの合計数
250 K=1
260 FOR I=1 TO N1
270 L=I
280 FOR J=1 TO N3
290 Wave_form(L)=Samp(K)
300 K=K+1
310 L=L+Inc
320 NEXT J
330 NEXT I
340 FOR I=N1+l TO N1+N2
350 L=I
360 FOR J=1 TO N3-1
370 Wave_form(L)=Samp(K)
380 K=K+ 1
390 L=L+Inc
400 NEXT J
410 NEXT I
420 END
```

SSPARM?

サブサンプリング・パラメータの問合せ。サンプルがGPIB出力バッファに直接送られる場合に、サブサンプリング(SSACまたはSSDCコマンド)された波形の再構成に必要なパラメータを返します(サンプルが読み取り値メモリに直接送られる場合は、再構成は自動です)。

SSPARM?により返される1番目のパラメータは、*N*個のサンプルを含むバーストの個数です。2番目のパラメータは、*N-1*個のサンプルを含むバーストの個数です。返される3番目のパラメータは、*N*の値です。例えば、10kHz信号をサブサンプリングし、5µsの*effective_interval*を使用して22個のサンプルを指定すると仮定します。この例では、マルチメータが合計4つのバーストを使用する必要があります。6個のサンプルが含まれるバーストが2つと、それぞれ5個のサンプルが含まれる2つのバーストです。したがって、SSPARM?により返される値は、2、2および6になります。

構文 **SSPARM?**

備考 • **関連コマンド**: SSAC、SSDC、SSRC、SWEEP

例 前のページのSSDCの例を参照してください。

SSRC

同期信号源。サブサンプリング(SSACまたはSSDCコマンド)では、SSRCコマンドを使用

すれば、バーストを外部信号または入力信号の電圧レベルに同期することができます。

同期ACVまたはACDCV(SETACV SYNCコマンド)では、SSRCコマンドを使用すれば、サンプリングを外部信号に同期することができます。HOLDパラメータを使用して、レベル・トリガが一定の時間リミット内に発生しなかった場合でも、測定法がランダム法へ変更されることを防げます。時間リミットは、AC帯域幅(ACBANDコマンド)設定で決まります。

構文 **SSRC [*source*][,*mode*]**

source

*source*パラメータの選択肢は次のとおりです。

source パラメータ	数値問合せの等価値	説明
EXT	2	裏面パネルの**Ext Trig**コネクタの外部入力に同期します。
LEVEL *	7	SLOPEコマンドによって指定されたスロープを使用して入力信号の電圧レベル(LEVELコマンド)に同期します。

* 同期ACVまたはACDCVでは、レベル・トリガ電圧(LEVELコマンド)とスロープ(SLOPEコマンド)は自動的に決定され、指定することはできません。

電源投入時の***source*** = LEVEL
デフォルトの***source*** = LEVEL

mode

*mode*パラメータは、同期ACVまたはACDCVが対象です。次の選択肢があります。

mode パラメータ	数値問合せの等価値	説明
AUTO	1	レベル・トリガ(デフォルト・モード)を使用した同期ACまたはACDCV(SETACV SYNC)の場合、読み取り中に入力信号が取り除かれ、一定の時間復帰しなかった場合、読み取りが完了するように測定法はランダム法へ変更されます(読み取り後、測定法はSYNCに戻ります)。
HOLD	4	入力信号が取り除かれても、測定法は同期からランダムへ自動的に変わりません。

* 同期ACまたはACDCVの時間リミットは、ACBANDコマンドを使用して指定した帯域幅によって決まります。

電源投入時の***mode*** = AUTO
デフォルトの***mode*** = AUTO

備考

- サブサンプリングでは、トリガ・イベントとサンプル・イベントは無視されます。サブサンプリングに適用されるトリガリング・イベントは、トリガ・アーム・イベント(TARMコマンド)と同期信号源イベント(SSRCコマンド)だけです。同期ACVまたはACDCV 測定(SETACV SYNCコマンド)では、指定したトリガ・アーム・イベント(TARMコマンド)、トリガ・イベント(TRIGコマンド)およびサンプル・イベント(NRDGSコマンド)がすべて満足されていなければ、同期信号源イベントがサンプリングを開始できません。

- サブサンプリングおよび同期AC測定では、波形の2周期以上でサンプルのバーストが取得されます。同期信号源イベントは、これらのバーストと入力信号の周期を同期します(つまり、通常、同期信号源イベントは1周期あたり1回発生する必要があります)。

- **問合せコマンド**　SSRC? 問合せコマンドは、カンマで区切られた 2 つの応答を返します。1番目の応答は現在の**信号源**です。2番目の応答は、現在の**モード**です。詳細については、本章の始めにある「問合せコマンド」を参照してください。

- **関連コマンド**: LEVEL、LFILTER、SETACV SYNC、SLOPE、SSAC、SSDC

例　次のページのプログラムでは、ピーク値が5Vの10kHz信号をデジタイズするためにSSACコマンドが使用されています。SWEEPコマンドは、マルチメータに対して、2µsの*effective_interval* (Eff_int variable)で1000個のサンプル(Num_samples variable)を取得するように命令します。測定では、同期信号源イベントに対してデフォルトのレベル・トリガが使用されます(入力信号からのトリガ、0%、AC結合、正スロープ)。行120はSYNイベントを生成し、サンプルを直接コンピュータへ転送します。行240～410は、サブサンプルされたデータをソートして、合成波形を生成します。合成波形は、Wave_form配列に保存されます。

```
10 OPTION BASE 1                  !配列の番号付けが1から始まります。
20 INTEGER Num_samples,Inc,I,J,K,L!変数を宣言します。
30 Num_samples=lOOO               !サンプル数を指定します。
40 Eff_int=2.0E-6                 !実効間隔を指定します。
50 INTEGER Int_samp(1:1000) BUFFER!整数バッファを作成します。
60 ALLOCATE REAL Wave_form(1:Num_samples)!保存済みデータ用の配列を作成します。
70 DATA ALLOCATE REAL Samp(1:Num_samples)!サンプル用の配列を作成します。
80 ASSIGN @Dvm TO 722             !マルチメータ・アドレスを割り当てます。
90 ASSIGN @Int_samp TO BUFFER Int_samp(*)!バッファ I/Oパス名を割り当てます。
100 OUTPUT @Dvm;"PRESET FAST;LEVEL;SLOPE;SSRC LEVEL;SSDC 10"
101 !高速演算、TARM SYN、レベルSYNC信号源0V、正のスロープ
105 !(デフォルト値)サブサンプリング(SINT出力フォーマット)、10Vレンジ
110 OUTPUT @Dvm;"SWEEP ";Eff_int,Num_samples
115 !2µsの実効間隔、1000個のサンプル
120 TRANSFER @Dvm TO @Int_samp;WAIT!SYNイベント、読み取り値を整数配列に
121 !転送します。コンピュータの整数フォーマットはSINTと同じため、
125 !ここでは、データ変換は不要です(整数配列は必要です)。
```

```
130 OUTPUT @Dvm;"ISCALE?"          !SINTフォーマットのスケール・ファクタを問い合わせます。
140 ENTER @Dvm; S                  !スケール・ファクタを入力します。
150 OUTPUT @Dvm;"SSPARM?"          !サンサンプリング・パラメータを問い合わせます。
160 ENTER @Dvm;N1,N2,N3            !サンサンプリング・パラメータを入力します。
170 FOR I=1 TO Num_samples
180 Samp(I)=Int_samp(I)            !個々の整数読み取り値を実数に変換します。
190 !フォーマット(次の行での整数オーバフローを防ぐために必要)
190 R=ABS(Samp(I))                 !絶対値を使用してOVLDがないか確認します。
200 IF R>=32767 THEN PRINT "OVLD" !OVLDの場合、過負荷メッセージを出力します。
210 Samp(I)=Samp(I)*S              !読み取り値にスケール・ファクタを掛けます。
220 Samp(I)=DROUND(Samp(I),4)      !ROUND TO 4 DIGITS
230 NEXT I
235 !--------------------------SORT SAMPLES------------------------------
240 Inc=N1+N2                      !バーストの総数
250 K=1
260 FOR I=1 TO N1
270 L=1
280 FOR J=1 TO N3
290 Wave_form(L)=Samp(K)
300 K=K+1
310 L=L+Inc
320 NEXT J
330 NEXT I
340 FOR I=N1+l TO N1+N2
350 L=I
360 FOR J=1 TO N3-1
370 Wave_form(L)=Samp(K)
380 K=K+ 1
390 L=L+Inc
400 NEXT J
410 NEXT I
420 END
```

次のプログラムでは、SSRC EXTイベントが同期AC電圧測定とともに使用されています。トリガ・イベントの発生後(トリガ・アーム・イベントとサンプル・イベントはAUTOです)、**Ext Trig**コネクタの最初のTTLの立ち下がりで、最初のバーストが開始されます。次に、必要なバースト数に達するまで、続く個々の外部トリガによりバーストが開始されます。

```
10 OUTPUT 722;"PRESET NORM"     !TARM AUTO、TRIG SYN、NRDGS 1、AUTO
20 OUTPUT 722;"ACV 10"          !AC電圧、10Vレンジ
30 OUTPUT 722;"SETACV SYNC"     !同期法
40 OUTPUT 722;"SSRC EXT"        !外部SYNC信号源・イベント
50 ENTER 722;A                  !読み取りをトリガし(TRIG SYN)、読み取り値を入力します。
60 PRINT A                      !読み取り値を出力します。
70 END
```

SSTATE

ステートの保存。マルチメータの現在のステートを保存して、これに名前を割り当てます。ステートは、RSTATEコマンドを使用して呼び出します。

構文 **SSTATE *name***

name

ステート名。ステート名の最大文字数は10です。名前には、英字、英数字、または0～127の整数が使用できます。英数名を使用する場合は、1文字目は英字でなければなりません。また、英字または英数字のステート名は、マルチメータのコマンドあるいはパラメータ、または保存済みサブプログラムと同じであってはいけません。文字_および?も、英字名または英数字名に使用できます。

整数のステート名(0～127)を使用すると､そのステートが保存される際にプレフィックス*STATE*がこの整数に割り当てられます。これにより、整数のステート名と整数のサブプログラム名が区別されます。例えば、8という名前で保存したステートは、*STATE8*として記録されます。このステートは、後で名前8または*STATE8*のいずれかを使用すれば呼び出せます。ステート0は、マルチメータの電源投入時のステート用に予約されています(下の1番目の備考を参照)。

電源投入時の*name* = なし
デフォルトの*name* = なし、パラメータが必要

備考

- マルチメータの電源が切断されると、現在のステートがステート0に保存されます。停電後にRSTATE 0を実行することで、マルチメータを前のステートに構成できます。
- すべてのステートが不揮発性メモリに保存されます(電源が切断されても失われません)。
- サブプログラム、読み取り値メモリの内容、および前面パネルのメニュー・モードは、保存済みステートの一部には含まれません。次の演算レジスタの内容は、ステートを保存すると保存されます(他の演算レジスタはすべて0に設定されます)。

DEGREE	REF
LOWER	RES
OFFSET	SCALE
PERC	UPPER

- マルチメータは、14kバイトのステート・メモリを備えています。1つのステートが約300バイトを占め、最大で46個のステートを保存できます。ステート0は、電源が切断されたときのマルチメータのステートを保存するために予約されています。ステート0は他のステートの保存にも使用できますが、保存したステートは、電源が切断されると現在のステートで上書きされます。
- **Recall State**キーを押して、上および下矢印キーを使用することで、前面パネルからすべての保存済みステートの名前を表示することができます。必要なステートが見つかったら、**Enter**キーを押してそのステートを呼び出します。

- **関連コマンド**: MSIZE、PURGE、RSTATE、SCRATCH

例
```
OUTPUT 722;"SSTATE B2 "          !B2という名前で現在のステートを保存します。
```

STB?

ステータス・バイトの問合せ。ステータス・レジスタには、マルチメータの様々な条件をモニタする7つのビットが含まれています。ある条件が発生すると、ステータス・レジスタ内の対応するビットが設定されます。STB?(ステータス・バイト?)コマンドは、設定されているビットを表す数値を返します。返された数値は、設定されたすべてのビットの重み和です。

構文 STB?

ステータス・レジスタ条件 ステータス・レジスタ条件とその重みは次のとおりです。

10進重み	ビット番号	ステータス・レジスタ条件
1	0	サブプログラム実行完了
2	1	上限または下限超過
4	2	SRQコマンド実行
8	3	電源投入
16	4	命令準備完了
32	5	エラー (エラー・レジスタを確認)
64	6	サービス要求(このビットはオフにできません)
128	7	データ使用可能

備考
- STB?コマンドを実行すると、マルチメータがSTB?コマンドの処理中であるために、レディ・ビット(ビット4)は常にクリア(レディでない)されます。

- CSBコマンドは、ステータス・レジスタをクリアします(ビット4、5および6は、そのビットを設定する条件がまだ存在する場合にはクリアされません)。RQSコマンドは、GPIBバスでどのステータス・レジスタ条件をアサートするのかを指定します。

- **関連コマンド**: CSB、EXTOUT、RQS、SPOLL (GPIB コマンド)

例
```
10 OUTPUT 722;"STB?"        !すべての設定されたビットの重み和を返します。
20 ENTER 722                !応答をコンピュータのA変数に入力します。
30 PRINT A                  !応答を出力します。
40 END
```

上のプログラムが重み和24を返したと仮定します。これは、重み値8(電源投入)と16(命令準備完了)が設定されていることを意味します。

SUB

サブプログラム。コマンド群を1つのサブプログラムとして保存し、サブプログラム名を割り当てます。

構文 **SUB** *name*

name

サブプログラム名。サブプログラム名の最大文字数は10です。英字、英数字、または0〜127の範囲の整数が使用できます。英数字名を使用する場合は、1文字目は英字でなければなりません。また、英字または英数字のサブプログラム名は、マルチメータのコマンドあるいはパラメータ、または保存済みステートと同じであってはいけません。文字_および?も、英字名または英数字名に使用できます。

整数のサブプログラム名(0〜127)を使用すると､そのサブプログラム保存される際にプレフィックス*SUB*がこの整数に割り当てられます。これにより、整数のサブプログラム名と整数のステート名が区別されます。例えば、*15*という名前で保存したサブプログラムは、*SUB15*として記録されます。このサブプログラムには、後で名前*15*または*SUB15*を使用すればアクセスできます。0(ゼロ)という名前のサブプログラムは、オートスタート・サブプログラムとして指定されます(以下の7番目の備考を参照)。

電源投入時の*name* = なし
デフォルトの*name* = なし、パラメータが必要

備考

- サブプログラム・エントリは、SUBEND コマンドで終わります。サブプログラムの実行にはCALLコマンドを使用し、サブプログラムの中断と再開には、それぞれPAUSEコマンドとCONTコマンドを使用します。
- 新しいサブプログラムを既存のサブプログラム名を使用して保存すると、新しいサブプログラムが古いプログラムを上書き(置換)します。
- サブプログラムを前面パネルから入力(保存)することはお勧めできません。前面パネル・ユーティリティ(上矢印および下矢印など)が、誤ってサブプログラム内に保存されることがあるためです。SUBコマンドを前面パネルから実行すると、SUBENDコマンドが実行されるか､RESETキーが押されるまで､ディスプレイに**SUB ENTRY MODE**と表示されます。サブプログラムを保存するまで、SUBENDコマンドは前面パネル・メニューには表示されません。
- SCRATCH、DELSUB、2番目のSUBコマンド、またはGPIB Device Clearコマンドがサブプログラム内で現れると、マルチメータはそのコマンドを保存せず、残りのサブプログラムを保存します。RESETコマンドが現れると、サブプログラムの実行が中断されます(サブプログラム内にRESETを保存しないでください)。
- サブプログラム/ステート・メモリの残りが800バイト未満の場合、サブプログラムは保存できません。
- エラーが検出された場合、またはGPIB Device Clearコマンドが受信された場合、サブプログラムの実行は中断されます。また、GPIB Device Clearコマンドは、サブプログラムの保存プロセスを中断します。

- サブプログラム内で読み取りを行う唯一の方法は、TARM　SGLコマンドまたはTRIG SGLコマンドを使用することです。これらのコマンドのいずれかが現れると、指定された読み取りがすべて行われるまで、マルチメータはサブプログラム内の次のコマンドを実行しません(これは、すべての構成コマンドとその他のトリガ・コマンドが、TARM SGLコマンドまたはTRIG SGLコマンドの前になければばならないということでもあります)。他のトリガ・アーム・イベントまたはトリガ・イベント(TARM EXTは除く。以下の備考を参照)はサブプログラム内で実行されますが、サブプログラムが完了するまで読み取りは開始されません。

- TARM EXTコマンドがサブプログラム内に現れると、常に、マルチメータはその**Ext Trig**コネクタで外部トリガが受信されるのを待って、サブプログラムの次の行を実行します。このため、サブプログラムの実行を外部機器と同期することができます。

- *0*という名前のサブプログラムは、マルチメータが電源投入時のシーケンスを終了すると、常に自動的に実行されます。これは、停電の後にマルチメータの前のステート(RSTATE 0)を呼び出すのに便利です。

- サブプログラムは、不揮発性メモリに保存されます(電源が切断されても失われません)。ただし、サブプログラムを圧縮すれば(COMPRESSコマンド)、サブプログラムは不揮発性メモリから移動され、電源が切断されると破棄されます。

- 関連コマンド: CALL、COMPRESS、CONT、DELSUB、PAUSE、SCRATCH、SUBEND

例

```
10 OPTION BASE 1                 !配列の番号付けが1から始まります。
20 DIM RDGS(5)                   !5個の読み取り値用の配列を宣言します。
30 OUTPUT 722;"SUB DCCUR2"       !DCCUR2という名前の次から行を保存します。
40 OUTPUT 722;"PRESET NORM"      !プリセットします。
50 OUTPUT 722;"MEM FIFO"         !読み取り値メモリをFIFOモードでオンにします。
60 OUTPUT 722;"DCV, 10, .01"     !DC電圧、10Vレンジ、分解能0.01%
70 OUTPUT 722; "NRDGS,5,AUTO"    !1回のトリガあたり5個の読み取り値
80 OUTPUT 722;"TRIG SGL"         !シングル・トリガ・モードを指定します。
90 OUTPUT 722; "SUBEND"          !サブプログラム保存の終わりを示します。
100 OUTPUT 722;"DISP MSG 'CALLING SUBPROGRAM'"
110 OUTPUT 722; "CALL DCCUR2"
120 ENTER 722;Rdgs(*)
130 PRINT Rdgs(*)
140 END
```

次のサブプログラム(CALL EXTPACE)が呼び出されると、TARM EXT(行70)が現れるまで、マルチメータがこのサブプログラムを1行ずつ実行します。その後、外部トリガが発生するまで、サブプログラムの実行は停止します。これにより、サブプログラムの実行を何らかの外部イベントと同期することができます。1番目の外部トリガが受信されると、サブプログラムの実行が再開します。次の行が現れると(TRIG SGL)、1000回の読み取りが行われるまでサブプログラムの実行は停止します。この読み取り後、サブプログラムにより測定ファンクションは2端子抵抗に、読み取り値の個数は100に変更されます。2番目のTARM EXTコマンドが現れると(行100)、サブプログラムの実行は別の外部トリガが発生するまで停止します。外部トリガが受信されると、TRIG SGLコマンドが現れ、これによって100回の読み取りが行われるまでサブプログラムが中断されます。読み取りが終わると、メッセージ**TEST FINISHED**が表示されます。

```
10 OUTPUT 722; "SUB EXTPACE"      !20～110行をプログラムとして保存します。
20 OUTPUT 722; "PRESET NORM"      !プリセット、読み取りの中断
30 OUTPUT 722; "MEM FIFO"         !読み取り値メモリをFIFOモードでオンします。
40 OUTPUT 722; "DCV 10"           !DC電圧測定、10Vレンジ
50 OUTPUT 722; "NRDGS 1000, AUTO" !1000個の読み取り値/トリガ、AUTOサンプル・イベント
60 OUTPUT 722; "TARM EXT"         !外部トリガ・アーム・イベント
70 OUTPUT 722; "TRIG SGL"         !シングル・トリガ・イベント
80 OUTPUT 722; "OHM 1E3"          !2端子抵抗、1kΩ レンジ
90 OUTPUT 722; "NRDGS 100, AUTO"  !100個の読み取り値/トリガ、AUTOサンプル・イベント
100 OUTPUT 722;"TARM EXT"         !外部トリガ・アーム・イベント
110 OUTPUT 722;"TRIG SGL"         !シングル・トリガ・イベント
120 OUTPUT 722;"DISP MSG,'TEST FINISHED'"!サブプログラムの終了を示します。
130 OUTPUT 722;"SUBEND"
140 END
```

SUBEND

サブプログラムの終わり。サブプログラムの終わりを示します。

構文 **SUBEND**

備考
- サブプログラを保存する場合は、SUBEND がサブプログラムの終わりを示します。サブプログラムが実行されると、SUBENDがステータス・レジスタのサブプログラムの完了を示すビット1を設定します(オンの場合)。
- **関連コマンド**: CALL、COMPRESS、CONT、DELSUB、PAUSE、SCRATCH、SUB

例 前のページのSUBの例を参照してください。

SWEEP

SWEEPコマンドは、サンプル(1回のトリガ・イベントあたりに(ほとんどの測定ファンクション)取得されるサンプル(読み取り値とサンプル総数)または1回のトリガアーム・イベントあたりに(サブサンプリングのみ)取得されるサンプル(読み取り値とサンプル総数)間の*effective_interval*を指定します。

構文 **SWEEP [*effective_interval*] [,*#_samples*]**

effective_interval

サブサンプリング(SSACまたはSSDC)の場合、このパラメータは再構成される波形のサンプルの間隔を指定します(詳細については第5章を参照)。他のすべての測定ファンクションでは、このパラメータはあるサンプルから次のサンプルまでの実際の時間間隔を指定します。サブサンプリングの場合、パラメータの有効範囲は、10ns刻みで10E-9～6000sです。他のすべての測定ファンクションでは、範囲は100ns刻みで(1/最大読み取り速度)～6000sです。

電源投入時の*effective_interval* = 100E-9
デフォルトの*effective_interval* = 20µs

#_samples

取得されるサンプル数を指定します。このパラメータの有効範囲は1～1.67E+7です。

電源投入時の*#_samples* = 1024
デフォルトの*#_samples* = 1024

備考

- DC電圧測定の最小実効間隔は10µsです。直接サンプリングの場合は20µs、サブサンプリングの場合は10nsです。

- SWEEPコマンドを使用して、NRDGS *n*,TIMERコマンドとTIMERコマンドを交換することができます。SWEEPとNRDGSには互換性があります。マルチメータは、プログラミングで最後に実行されたコマンドを使用します。SWEEPコマンドを実行すると、サンプル・イベントが自動的にTIMERに設定されます。電源投入時、RESETステートまたはPRESETステートでは、マルチメータはNRDGSコマンドを使用します。SWEEPの電源投入時の値は､サブサンプリングでのみ使用できます(NRDGSはサブサンプリングに適用されないため)。

- SWEEP関数またはTIMER関数は、同期法またはランダム法を使用したACあるいはAC+DC電圧測定(SETACV SYNCまたはRNDM)、または周波数／周期測定には使用できません。

- SWEEPコマンド(TIMERイベント)を使用すると、オートレンジングは中断されます(通常、SWEEPを使用するときは固定レンジを使用してください)。

- **問合せコマンド**　SWEEP?問合せコマンドは、カンマで区切られた2つの応答を返します。1番目は、指定された*effective_interval*です。2番目は、指定された*#_samples*です。詳細については、本章の始めにある「問合せコマンド」を参照してください。

- **関連コマンド**: FUNC、NRDGS、TIMER

例

以下のページのプログラムでは、SSACコマンドがピーク値5Vの10kHz信号のデジタイズに使用されています。SWEEPコマンドは、マルチメータに対して、2µsの*effective_interval* (Eff_int variable)で1000個のサンプル(Num_samples変数)を取得するように命令します。測定では、同期信号源イベントに対してデフォルトのレベル・トリガが使用されます(入力信号からのトリガ、0%、AC結合、正のスロープ)。

行120は、SYNイベントを生成して、サンプルを直接コンピュータへ転送します。行240～410は、サブサンプリングされたデータをソートして合成波形を生成します。合成波形は、Wave_form配列に保存されます。

```
10 OPTION BASE 1                    !配列の番号付けは1から始まります。
20 INTEGER Num_samples,Inc,I,J,K,L!変数を宣言します。
30 Num_samples=lOOO                 !サンプル数を指定します。
40 Eff_int=2.0E-6                   !実効間隔を指定します。
50 INTEGER Int_samp(1:1000) BUFFER!整数バッファを作成します。
60 ALLOCATE REAL Wave_form(1:Num_samples)!ソートされたデータ用の配列を作成します。
```

```
70 ALLOCATE REAL Samp(1:Num_samples)!サンプル用の配列を作成します。
80 ASSIGN @Dvm TO 722              !マルチメータ・アドレスを割り当てます。
90 ASSIGN @Int_samp TO BUFFER Int_samp(*)!バッファ I/Oパス名を割り当てます。
100 OUTPUT @Dvm;"PRESET FAST;LEVEL;SLOPE;SSRC LEVEL;SSDC 10"
101 !高速演算、TARM SYN、レベルSYNC信号源0V、正のスロープ
105 !(デフォルト値)サブサンプリング(SINT出力フォーマット)、10Vレンジ
110 OUTPUT @Dvm;"SWEEP ";Eff_int,Num_samples
115 !2µsの実効間隔、1000個のサンプル
120 TRANSFER @Dvm TO @Int_samp;WAIT!SYN EVENT,TRANSFER READINGS INTO
121 !整数配列。コンピュータの整数フォーマットがSINTと同じため、
125 !ここでは、データ変換は不要です(整数配列は必要)。
130 OUTPUT @Dvm;"ISCALE?"          !SINTフォーマットのスケール・ファクタを問い合わせます。
140 ENTER @Dvm; S                  !スケール・ファクタを入力します。
150 OUTPUT @Dvm;"SSPARM?"          !サンサンプリング・パラメータを問い合わせます。
160 ENTER @Dvm;N1,N2,N3            !サンサンプリング・パラメータを入力します。
170 FOR I=1 TO Num_samples
180 Samp(I)=lnt_samp(I)            !個々の整数読み取り値を実数フォーマットに変換
190 !します(次の行での整数のオーバフローを防ぐために必要)。
190 R=ABS(Samp(I))                 !絶対値を使用してOVLDがないか確認します。
200 IF R>=32767 THEN PRINT "OVLD" !OVLDの場合、過負荷メッセージを出力します。
210 Samp(I)=Samp(I)*S              !読み取り値にスケール・ファクタを掛けます。
220 Samp(I)=DROUND(Samp(I),4)      !4桁に丸めます。
230 NEXT I
235 !--------------------------SORT SAMPLES------------------------------
240 Inc=N1+N2                      !バーストの総数
250 K=1
260 FOR I=1 TO N1
270 L=1
280 FOR J=1 TO N3
290 Wave_form(L)=Samp(K)
300 K=K+1
310 L=L+Inc
320 NEXT J
330 NEXT I
340 FOR I=N1+l TO N1+N2
350 L=I
360 FOR J=1 TO N3-1
370 Wave_form(L)=Samp(K)
380 K=K+1
390 L=L+Inc
400 NEXT J
410 NEXT I
420 END
```

T

Tは、TRIGコマンドの省略形です。

構文 **T [*event*]**

詳細については、TRIGコマンドを参照してください。

TARM

トリガ・アーム。トリガ・イベント(TRIG コマンド)をオン(アーム)するイベントを定義します。このコマンドは、複数の測定サイクルの実行にも使用できます。

構文 **TARM [*event*][,*number_arms*]**

event

*event*パラメータの選択肢は次のとおりです。

event パラメータ	数値 問合せの 等価値	説明
AUTO	1	常にアーミング状態
EXT	2	Ext TrigコネクタのTTL立ち下がり後にアーミングします(TARM EXTを実行すると、TBUFFがONの場合にトリガ・バッファがクリアされます)。
SGL	3	1回アーミングし(TARM SGLの受信時)、HOLDになります。
HOLD	4	トリガをオフにします。
SYN	5	マルチメータの出力バッファが空、読み取り値メモリがオフあるいは空、かつコントローラがデータを要求したときにアーミングします。

電源投入時の*event* = AUTO
デフォルトの*event* = AUTO

number_arms

*number_arms*パラメータは、SGLトリガ・アーム・イベントとともに使用された場合にのみ有効です。この場合、有効範囲は0～2.1E+9です。SGLイベントを使用して0または1を指定すると、デフォルト値(1)を使用した場合と同じになりです。トリガが1回アーミングされて、HOLDステート(オフ)に戻ります。1より大きな数値を*number_arms*パラメータとして指定すると、「複数のアーミング」を選択したことになります。複数のアーミングでは、*number_arms*パラメータを満たすのに十分なシングル・トリガをマルチメータが発生します。詳細については、下の備考の「複数のアーミング」を参照してください。

電源投入時の*number_arms* = 1 (複数のアーミングはオフ)
デフォルトの*number_arms* = 1 (複数のアーミングはオフ)

備考

- サブサンプリング(第5章を参照)を除くすべての測定ファンクションでは、トリガ・アーム・イベントはトリガ・イベント(TRIGコマンド)とサンプル・イベント(NRDGSまたはSWEEPコマンド)とともに動作します。測定を行うには、最初にトリガ・アーム・イベントが発生し、次にトリガ・イベント、最後にサンプル・イベントが発生する必要があります。

- トリガ・アーム・イベントは、必ずしもマルチメータをトリガしません。トリガ・イベントをオンにして、マルチメータのトリガ・イベントへの応答を可能にするだけです。各種イベントの相互関係の詳細については、第4章の「トリガ」を参照してください。

- 複数のアーミング: 複数のアーミングを使用する場合は、トリガ・アーム・イベントがSGLとして指定されていなければなりません。複数のアーミングを指定するTARMコマンドをマルチメータが実行すると、すべての測定サイクルが完了するまでマルチメータがGPIBバスをホールドします。例えば、*number_arms*を5、1サイクルあたり10個の読み取り値(NRDGSコマンド)を指定した場合、読み取り値が10個の測定サイクルが5回行われます。これによりバスがホールドされるため、TARMコマンドはプログラムの最終行に置く必要があり、同期トリガ・イベントまたはサンプル・イベントは使用できません。

- **問合せコマンド**　TARM?問合せコマンドは、現在選択されているトリガ・アーム・イベントを返します。詳細については、本章の始めにある「問合せコマンド」を参照してください。

- **関連コマンド**: NRDGS、SWEEP、TRIG

例

```
OUTPUT 722; "TARM AUTO, 0"      !オート・トリガ・アーミング(常にアーミング)

10 OUTPUT 722; "TARM HOLD"      !測定を中断します。
20 OUTPUT 722; "OHM"            !2端子抵抗測定を測定します。
30 OUTPUT 722; "MEM FIFO        !読み取り値メモリをFIFOモードでオンにします。
40 OUTPUT 722; "NRDGS 5"        !1つのサンプル・イベントあたり5個の読み取り値(AUTO)
50 OUTPUT 722; "TARM SGL"       !1つの連続測定をオンにします。
60 END

10 OUTPUT 722; "DCV"            !DC電圧測定を選択します。
20 OUTPUT 722; "TARM HOLD"      !測定を中断します。
30 OUTPUT 722; "TRIG AUTO"      !AUTOをトリガ・イベントとして選択します。
40 OUTPUT 722; "MEM FIFO"       !読み取り値メモリをFIFOモードでオンにします。
50 OUTPUT 722; "NRDGS 3, AUTO"  !1つのサンプル・イベントあたり3個の読み取り値(AUTO)
60 OUTPUT 722; "TARM SGL,5"     !5サイクルに対して複数のアーミングを選択します。
70 END
```

このプログラムでは、行60が各測定サイクルに対してトリガを1回アーミングします。これが5回行われます。5番目のサイクルの後、トリガ・アーミングはHOLDに戻ります。このプログラムは、15個の読み取り値を(1回のトリガ・イベントあたり3個の読み取り値で5回)読み取り値メモリに入れます。

入力バッファがオンされていない場合、行60が、すべての測定サイクルが完了するまでGPIBバスをホールドします。バスの制御をすぐに戻したい場合は、行60を次の行に置き

換えてcr lfを抑制します。

```
60 OUTPUT 722 USING "#,K"TARM SGL, 5;"
```

上の行では、#イメージ指示子が、*cr lf*を抑制します。Kイメージ指示子は、前後のスペースを抑制し、コマンドを自由フィールド・フォーマットで出力します。TARM SGL.5に続くセミコロンに注意してください。これは、コマンドの終わりをマルチメータに対して示し、*cr lf*を抑制する場合にはこれが必要です。

TBUFF

トリガ・バッファ。マルチメータの外部トリガ・バッファをオンまたはオフにします。

構文 **TBUFF [*control*]**

control

*control*パラメータの選択肢は次のとおりです。

control パラメータ	数値問合せの 等価値	説明
OFF	0	トリガ・バッファをオフにし、これによってTRIGGER TOO FASTエラーがオンになります。
ON	1	トリガ・バッファをオンにしてクリアし、これによりTRIGGER TOO FASTエラーがオフになります。

電源投入時の***control*** = OFF
デフォルトの***control*** = OFF

備考

- TBUFFをONにすることで、外部EXTトリガ・アーム・イベント、トリガ・イベントまたはサンプル・イベントを使用する際に発生する可能性のあるTRIGGER TOO FASTエラーが修正されます。TBUFFがOFFの場合、読み取り中に外部トリガが生じると、TRIGGER TOO FASTエラーが生成されて、そのトリガは無視されます。TBUFFがONの場合は、読み取り中に生じた最初の外部トリガが保存され、このトリガでも後続のトリガでもエラーは生成されません。読み取りが完了すると、保存されたトリガがEXTイベントを満たしますが、これはマルチメータがそのようにプログラムされている場合です。

- RESETコマンドを実行すると、TBUFFがOFFに設定されます。

- **問合せコマンド**　TBUFF?問合せコマンドは、現在のトリガ・バッファリング・モードを返します。詳細については、本章の始めにある「問合せコマンド」を参照してください。

- **関連コマンド**: EXTOUT、NRDGS、TRIG

例

```
OUTPUT 722;"TBUFF ON" !TRIGGER TOO FASTエラーをオフにします。
```

TEMP?

温度の問合せ。マルチメータの°C単位の内部温度を返します。

構文 **TEMP?**

備考

- マルチメータの温度をモニタすると、いつ自動構成を実行するかの決定に役立ちます。
- **関連コマンド**: ACAL, CAL, CALSTR

例

```
10 OUTPUT 722; "TEMP?"        !温度を読み取ります。
20 ENTER 722; A               !結果を入力します。
30 PRINT A                    !結果を出力します。
40 END
```

TERM

前のマルチメータでは、TERMコマンドはマルチメータの入力端子を内部的に接続または切断していました。3458は言語の互換性を維持するためにTERMコマンドを受け付けますが、入力端子はリモートから制御できないため、応答しません。

構文 **TERM [*source*]**

source

*source*パラメータの選択肢は次のとおりです。

source **パラメータ**	**数値問合せの等価値**	**説明**
OPEN	0	エラー・メッセージを生成します。
FRONT	1	**Terminals**スイッチが**Rear**に設定されるとエラー・メッセージを生成します。
REAR	2	**Terminals**スイッチが**Front**に設定されるとエラー・メッセージを生成します。

電源投入時の*source* = なし
デフォルトの *source* = FRONT

備考

- **問合せコマンド**　TERM?問合せコマンドは、前面パネルの**Terminals**スイッチにより選択されている入力端子(FRONTまたはREAR)を示す応答を返します。

TEST

マルチメータに一連の内部セルフテストを実行させます。

構文 **TEST**

備考
- 入力信号をすべて切り離した後でセルフテストを実行してください。入力信号をマルチメータに接続したままにしておくと、セルフテスト不良が起こる場合があります。

- ハードウェア・エラーが検出された場合、マルチメータがエラー・レジスタのビット0と、補助エラー・レジスタの詳細なビットを設定します。エラー・レジスタのビットが設定されている場合には、常にディスプレイの**ERR**インジケータが点灯します。エラー・レジスタには、ERRSTR?(両方のレジスタ)、ERR?(エラー・レジスタのみ)またはAUXERR?(補助エラー・レジスタのみ)を使用してアクセスできます。

- **関連コマンド**: AUXERR?、ERR?、ERRSTR?

例
```
OUTPUT 722;."TEST" !セルフテストを実行します。
```

TIMER

TIMERコマンドは、NRDGSコマンドのTIMERサンプル・イベントの時間間隔を定義します。TIMERイベントを使用すると、読み取りの間に時間間隔が挿入されます。

構文 **TIMER [*time*]**

time

*time*パラメータの有効範囲は、100ns刻みで(1/最大サンプリング・レート)〜6000sです。

電源投入時の*time* = 1s
デフォルトの*time* = 1s

備考
- TIMER イベントを使用する場合、最初の読み取りは時間間隔を置かずに行われます。DELAYコマンドを使用すれば、最初の読み取りの前に時間間隔を挿入できます。

- TIMERイベントを使用する場合、オートレンジングは中断されます(通常、TIMERイベントを使用する場合には固定レンジを選択してください)。TIMERサンプル・イベントを指定した際にオートレンジングがオンになっていた場合、オートレンジングは、別のサンプル・イベントを指定すると再開されます。

- SWEEPコマンドは、任意の測定ファンクションで、2つのコマンド、NRDGS*n*、TIMERおよびTIMER *n*を交換するために使用できます。SWEEPとNRDGSには互換性があります。マルチメータはプログラミング内で最後に実行されたコマンドを使用します。SWEEPコマンドを実行すると、サンプル・イベントが自動的にTIMERに設定されます。電源投入時、RESETまたはPRESETステートでは、マルチメータがNRDGSコマンドを使用します。SWEEPの電源投入時の値は、サブサンプリングにのみ使用できます(NRDGSはサブサンプリングに適用されないため)。

- TIMER(SWEEP)イベントは、同期法あるいはランダム法(SETACV SYNCまたはRNDM)を使用するACあるいはAC+DC電圧測定、または周波数／周期測定には使用できません。

- **問合せコマンド**　TIMER? 問合せコマンドは、NRDGS タイマ・イベント用の現在の時間間隔(秒単位)を返します。

- **関連コマンド**: DELAY、NRDGS、SWEEP

例
```
10 OUTPUT 722;"TRIG HOLD"       !測定を中断します。
20 OUTPUT 722;"INBUF ON"        !入力バッファをオンにします。
30 OUTPUT 722;"DCV 10"          !DC電圧、10Vレンジ
40 OUTPUT 722;"NPLC .1"         !積分時間として0.1 PLCを選択します。
50 OUTPUT 722;"AZERO OFF"       !オートゼロをオフにします。
60 OUTPUT 722;"MEM FIFO"        !読み取り値メモリ(FIFO モード)をオンにします。
70 OUTPUT 722;"TIMER 2"         !2秒間隔を選択します。
80 OUTPUT 722;"NRDGS 10 TIMER" !1つのサンプル・イベント(TIMER)あたり10個の読み取り値
90 OUTPUT 722;"TRIG SGL"        !1回トリガします。
100 END
```

TONE

マルチメータがビープ音を1回出します。次に、マルチメータは前のBEEPモード(OFFまたはONのいずれか)に戻ります。

構文 **TONE**

関連コマンド: BEEP

例
```
OUTPUT 722; "TONE"  ! ビープ音を出します。
```

TRIG

トリガ・イベントを指定します。

構文 **TRIG [*event*]**

event

*event*パラメータの選択肢は次のとおりです。

event パラメータ	**数値問合せの等価値**	**説明**
AUTO	1	マルチメータがビジーでない場合は、常にトリガします。
EXT	2	外部トリガ・コネクタの立ち下がりTTL信号でトリガします。
SGL	3	1回トリガして(TRIG SGLの受信時)、TRIG HOLDに戻ります。
HOLD	4	読み取りをオフにします。
SYN	5	マルチメータの出力バッファが空、読み取り値メモリがオフまたは空、かつコントローラがデータを要求した場合にトリガします。
LEVEL *	7	SLOPEコマンドにより指定されたスロープで、LEVELコマンドにより指定された電圧に入力信号が達するとトリガします。
LINE **	8	AC電源ライン電圧のゼロ交差でトリガします。

* LEVELトリガ・イベントは、DC電圧測定と直接サンプリングした測定でのみ使用できます。

** LINEトリガ・イベントは、サンプリングしたACあるいはAC+DC電圧の測定(SETACV RNDMあるいはSYNC)、または周波数／周期測定には使用できません。

電源投入時の*event* = AUTO

デフォルトの*event* = SGL

備考

- サブサンプリング(第5章を参照)を除くすべての測定では、トリガ・イベントは トリガ・アーム・イベント(TARMコマンド)とサンプル・イベント(NRDGSコマンド)とともに動作します(トリガ・イベントとサンプル・イベントは、サブサンプリングでは無視されます)。測定を行うためには、トリガ・アーム・イベントが最初に発生した後、トリガ・イベントが続き、最後にサンプル・イベントが発生しなくてはなりません。トリガ・イベントでは、測定は開始されません。測定をオンにして、測定を可能にするだけです。測定は、サンプル・イベント(NRDGSコマンドまたはSWEEPコマンド)が発生すると開始されます。ほとんどの測定ファンクションでの各種イベントの相互関係の詳細については、第4章の「測定のトリガ」を参照してください。サブサンプリングについては、第5章を参照してください。

- **問合せコマンド** TRIG? 問合せコマンドは、指定されているトリガ・イベント返します。詳細については、本章の始めにある「問合せコマンド」を参照してください。

- **関連コマンド**: LEVEL、LFILTER、NRDGS、SLOPE、SWEEP、T、TARM、TBUFF

例

```
OUTPUT 722; "TRIG AUTO"!SELECTS AUTO TRIGGER
```

次のプログラムは、マルチメータが正しく構成されるまで測定を中断する方法を示します。行20が、トリガ・イベントをHOLDに設定することで測定を中断します。行30と行40では､1回のトリガ･イベントあたり30個のDC電圧読み取り値を取得ための構成を行っています。行50は、マルチメータに30回の読み取りを行わせるシングル・トリガを生成します。読み取りの完了後、トリガ・イベントはHOLDへ戻ります。

```
10 OUTPUT 722;"ESET"            !電源投入時ステートに戻ります。
20 OUTPUT 722;"TRIG HOLD"       !読み取りを中断します。
30 OUTPUT 722;"DCV 10"          !DC電圧測定、10Vレンジ
40 OUTPUT 722;"NRDGS 30,AUTO"   !1つのサンプル・イベント(AUTO)あたり30個の読み取り値
50 OUTPUT 722;"TRIG SGL"        !シングル・トリガを生成します。
60 END
```

第7章　3458A用のBASIC言語

第7章　3458A用のBASIC言語

概要

本章では、3458Aの内部BASIC言語オペレーティング・システムによりサポートされているBASICコマンドについて説明します。この機能により、簡単なBASICサブプログラムを書いたりダウンロードしたりしてマルチメータの動作をカスタマイズすることで、数多くの特殊要件を容易に満たすことができます。次のような場合に、内部BASIC言語が役立ちます。

- 前面パネル･ディスプレイの表示値をカスタマイズして、より分かりやすくする。
- 新しい測定ファンクション、演算、または特殊なトランスジューサの線形化の追加
- スループットの極めて高いシステム測定を実行するためのマルチメータの構成
- GPIBを多用したマルチメータ内部でのデータ低減の実行
- FFT用などのMotorola 68000バイナリ・プログラムのダウンロード
- 特殊なニーズを満たすAgilentカスタム・バイナリ・プログラム

BASIC言語の操作法

マルチメータのSUBコマンドを使用して、3458Aのプログラム・メモリ・スペース内に新規サブプログラムを作成するだけです。第6章で説明したマルチメータ・コマンドを使用できます。また、本章で説明する新しいBASIC言語コマンドを含めて簡単なBASICプログラムを作成することもできます。**非常に簡単です。そして、これらのコマンドは、3458Aの測定器ファームウェアのすべてのリビジョン(特記事項ある場合は除く)で動作します。**

サブプログラムは、GPIBバスから呼び出したり、シングル・キー・プレス操作のために前面パネルのユーザ定義キー (F0からF9)に割り当てたり、他のサブプログラム内から呼び出すことができます。

3458AのBASIC言語は、以下はサポートしていません。

- 文字列変数および演算
- 行番号
- GOTO文
- GOSUB文

- ローカル変数(すべての変数がグローバルです)
- パラメータ引き渡し
- 本章に記載されていない他のBASICコマンド

BASIC言語コマンド

このセクションでは、3458Aの内部BASIC言語オペレーティング・システムによりサポートされているBASIC言語コマンドの概要を説明します。これらのコマンドの詳細と例については、本書の後のセクションを参照してください。

変数と配列

注記 すべての配列のインデックスは、0から*size*までです(OPTION BASE 0)。

LET *user_variable = expression*

REAL *variable_l, variable_2, . . .* 実数のユーザ変数を宣言します。実数配列を宣言するための*REAL variable_l (size)*も使用可能です。REALは、64ビット値です。

INTEGER *variable_l, variable_2, . . .* 整数のユーザ変数を宣言します。整数配列を宣言するための*INTEGER variable_1 (size)*も使用可能です。INTEGERは、16ビット値です。

DIM *array_name (size), . . .* 配列を宣言します。

FILL *array_name, list* 名前付き配列を、続く数値リストのデータで埋めます。埋められた配列は、**不揮発性**メモリ・スペースに保存されます。

サブプログラムでの変数使用における制限

ある変数を参照しているすべてのサブプログラムは、この変数を定義していなければなりません。変数の定義は、同じ変数名の他のすべての定義と一致していなければなりません。サブプログラムの間でユーザ変数の定義が異なっていると、電源を入れなおしたときに問題が生じる場合があります。これは、メモリを節約するためのサブプログラムの内部的な保存法によるためです。サブプログラムの実行可能コードは、実際にはマルチメータの電源投入時のルーチンの中で再構成されます。

FILLコマンドは注意して使用してください。電源を入れなおした場合、このコマンドは動作しません。その時点で、実質的にこのコマンドがサブプログラムから削除されるからです。割り当てられた各値に対して個別にLET文を使用してください。

演算

数値演算：+, -, *, /, ^
=, >, <, >=, <=, <>

```
               DIV, MOD, ABS, SQR, LOG, EXP, LGT, SIN, COS, ATN
バイナリ演算: AND, OR, EXOR, NOT, BINAND, BINCMP, BINEOR,
               BINIOR, BIT, ROTATE, SHIFT
```

サブプログラムの定義/削除

SUB *sub_name* サブプログラムがどこで始まるかを識別し、名前をサブプログラムに割り当てます。

SUBEND *sub_name* サブプログラムがどこで終わるかを識別し、サブプログラムのエントリを終了します。

DELSUB *sub_name* 指定されたサブプログラムを内部メモリから削除します。

SCRATCH すべての3458A サブプログラム、変数および配列を内部メモリから削除(消去)します。

CAT 現在内部メモリに保存されているすべての3458Aサブプログラム、単純変数、保存済みステートおよび配列の名前をリストします(400文字まで)。

LIST *sub_name* 指定したサブプログラムをリストします(400文字まで)。

COMPRESS *sub_name* 指定したサブプログラムのテキストをメモリから削除します。

サブプログラム実行コマンド

CALL *sub_name* 名前付きサブプログラムを実行し、完了するまで待って他のコマンドを実行します。

PAUSE CALLコマンドを使用して最後に実行されたサブプログラムを一時停止します。

CONT PAUSEコマンドの実行後にサブプログラムの実行を再開します。

ループと分岐

FOR *counter =initial_value* **TO** *final-value* [**STEP** *step_size*]
NEXT counter

WHILE *expression*
ENDWHILE

IF expression **THEN**
[**ELSE**]
ENDIF

バイナリ・プログラム

CALLARRAY *array name, integer_list* 指定した配列の内部アドレスをフェッチし、そこから実行を開始します。

この配列は、FILLコマンドを使用してデータ(ASCIIに変換される)がロードされていなければなりません。バイナリ・データは、相対アドレスを使用して書かれたMotorola 68000実行可能コードでなければなりません。

新しいマルチメータ コマンド

次のコマンドは第6章では説明していませんが、便宜上、本章に含めてあります。これらのコマンドは、3458Aの測定器ファームウェアのすべてのリビジョン(特記事項がある場合を除く)で動作します。

ENTER *user_variable* マルチメータの読み取り値メモリから読み値をユーザ変数へ転送します。マルチメータの読み取り値は、実行後に消去されます。例: ENTER Dmm

OUTPUT *user_variable* ユーザ変数の現在の値を出力します。データは、サブプログラムが実行されたソースに基づいて、ディスプレイまたはGPIB出力バッファのいずれかに送られます。 例: OUTPUT Result

U_RANGE 現在のファンクションで、レンジを1つ上げます。

D_RANGE 現在のファンクションで、レンジを1つ下げます。

DSP *string* or *user_variable* マルチメータの前面パネル・ディスプレイに、テキストとユーザ変数の両方を出力します(REV 2.1ファームウェア以上でのみ使用可能)。

DSP? 現在の前面パネル・ディスプレイを読み取ります。

SCROLL LEFT | RIGHT 現在の前面パネル・ディスプレイを1文字右または左にスクロールします。これは、DISPコマンドを使用して送られたテキストのみに適用されます(詳細については第6章を参照してください)。

ECHO *string* 指定した文字列を、マルチメータの前面パネル・ディスプレイまたはGPIBのいずれかにエコー・バックします。データは、サブプログラムが実行されたソースに基づいて、ディスプレイまたはGPIB出力バッファのいずれかに送られます。

RETURN サブプログラム内で、SUBEND文の前に戻るために使用します。

RMATHV *register* , *user_variable* 標準のマルチメータ演算レジスタをユーザ変数に読み込みます(REV 5.1ファームウェア以上でのみ使用可能です)。

WAIT *msec* 次のコマンドの実行を待ちます(最大32秒)。

3458A BASIC言語のサンプル・プログラム

次のサンプル・プログラムは、3458Aの内部BASIC言語と新しいマルチメータ・コマンドの使用方法を示したものです。このプログラム・サンプルでは、シリーズ300 BASICコンピュータを、プログラム作成と、GPIBインタフェースでのプログラムのマルチメータへのダウンロードに使用しています。マルチメータのバス・アドレスは22で、コンピュータのGPIBインタフェース・アドレスは700に設定されています。

```
 10 !
 20 ! このプログラムは、3458Aを使用して平均を計算します
 30 ! (最大値と最小値は捨てます)。
 40 ! 次の4つのBASIC言語コマンドが使用されています。
 50 ! RMATHV、LET、REALおよびOUTPUTです。
 60 !
 70 !
 80 ! RMATHV - 変数を演算レジスタの現在の値で埋めます。
 90 !                          RMATHと似ています。
100 !
110 ! OUTPUT - 値をコマンドが実行されたソースに返します
120 !                          (この例では、サブプログラムをGPIBバスから
130 !                          呼び出しているため、AVGの値はバスを使用して
140 !                          送られます)。
150 !
160 ! LETおよびREAL - 値を指定した変数に割り当てます。
170 !
180 !
190 !
200 DIM Rdgs(1:300)               !データ配列を宣言します。
210 ASSIGN @Dvm TO 722            !GPIBアドレスを設定します
220 !
230 CLEAR @Dvm
240 OUTPUT @ Dvm; "RESET"
250 WAIT 0.5
260 !
270 OUTPUT @Dvm; "PRESET FAST"
280 OUTPUT @Dvm; "OHM 1000"
290 OUTPUT @Dvm; "APER 167E-6"
300 OUTPUT @ Dvm; "OFORMAT ASCII"
310 OUTPUT @Dvm; "MEM FIFO"
320 OUTPUT @Dvm; "NRDGS 300,TIMER" !300個の読み取り値を取得するように設定します。
330 OUTPUT @Dvm; "TIMER 0.0002"   !サンプリング・レート5000 読み取り/s
340 !
350 !
360 !
370 OUTPUT @Dvm; "SUB CALC_MEAN"  !DMMサブプログラムの始まり
380 OUTPUT @Dvm; "REAL BIG,SMALL,AVG"!ユーザ変数を宣言します。
390 OUTPUT @Dvm; "MMATH STAT"
400 OUTPUT @Dvm; "RMATHV MEAN, AVG"!新しいDMMコマンド
410 OUTPUT @Dvm; "RMATHV UPPER, BIG"!新しいDMMコマンド
420 OUTPUT @Dvm; "RMATHV LOWER, SMALL"!新しいDMMコマンド
430 OUTPUT @Dvm; "LET M=(AVG*300-BIG-SMALL)/298"!Mを計算する式
440 OUTPUT @Dvm; "OUTPUT M"       !計算結果をバスへ送ります。
450 OUTPUT @Dvm; "SUBEND"         !DMMサブプログラムの終わり
460 OUTPUT @Dvm; "TARM SGL"       !DMM取り込みをトリガします。
470 T0=TIMEDATE                   !開始時間を保存します。
480 T1=TIMEDATE
490 OUTPUT @ Dvm; "CALL CALC_MEAN"!DMMにサブプログラムの実行を指示します。
```

```
500 ENTER @Dvm; Mean              !Mをコンピュータに取り込みます。
510 T2=TIMEDATE                   !終了時間を保存します。
520 PRINT"MEAN";Mean;"TRANSFER AND CALCULATION SPEED";T2-T1-(T1 -T0)
530 PRINT
540 END
```

プログラムを実行した場合のサンプル結果:

```
MEAN 54.73391112 TRANSFER AND CALCULATION SPEED .399963378906
```

変数と配列

3458Aは、2つの形式の数値変数を採用しています。単純変数("スカラ"とも呼ばれます)と添字付き配列です。3458Aでの変数の使い方は、拡張BASIC言語の場合とよく似ています。3458Aでは、文字列変数は**使えません**。変数はすべて、前面パネル、GPIBおよびサブプログラムの操作で共通で、グローバルです。このため、変数値を動的に変化させることが可能です。

型宣言

3458Aは、その変数に2つのデータ型を使用します。整数または実数です。変数はすべて、整数と宣言しなければ実数です。実数値の範囲は次のとおりです。

–1.797 693 134 862 315 X 10^{308} から1.797 693 134 862 315 X 10^{308}

使用可能な最小非ゼロ実数値は次のとおりです。

±2.225 073 858 507 202 X 10^{-308}

実数は、ゼロの値をとることができます。

整数は、以下の範囲の任意の整数値をとることができます。

–32767から +32767

DIMコマンドは、実数配列を宣言します。INTEGERコマンドは、整数変数または配列を宣言します。REALコマンドは、実数変数または配列を宣言します。

次のプログラム文は、10個の要素(番号は0〜9)を持つ実数配列Aを宣言しています。

```
OUTPUT 722; "DIM A(9)"
```

次のプログラム文は、10個の要素(番号は0〜9)を持つ整数配列IARRAYと、整数変数Bを宣言しています。

```
OUTPUT 722; "INTEGER IARRAY(9),B"
```

次のプログラム文は、10個の要素(番号は0〜9)を持つ実数配列RARRAYと、変数Cを宣言しています。

```
OUTPUT 722; "REAL RARRAY(9), C"
```

LETコマンドのある割り当て文の中に変数名が現れると、3458Aが自動的に変数を宣言します。例えば、次の文は指定した変数名を自動的に宣言します。

```
OUTPUT 722; "LET A=SIN(.223)"
OUTPUT 722; "LET B=3.14159"
```

一部の3458Aコマンドは、パラメータに対して変数を定義する際に特定の変数を期待します。例えば、TIMEコマンドは、実数を期待とします。同様に、数値結果を返すコマンドは、特定の数値タイプを返します。LINE?コマンドは、整数値を返します。返される測定値は、実数値です。すべての変数は、別に指定しない限りREALです。

注記

プログラムミングのヒント
一度配列タイプを宣言したら、先にメモリをスクラッチしないと(第6章の「SCRATCHコマンド」を参照)、これを別のタイプとして再度宣言することはできません。整数を期待するコマンドが実数を参照すると、3458Aが実数を整数に変換します。同様に、実数を期待するコマンドが整数を参照すると、3458Aが整数を実数に変換します。したがって、変数をその用途にしたがって割り当てることで、システム・オーバヘッド時間を最小限に抑えることができます。例は、次のとおりです。
OUTPUT 722; "REAL TIME_INT; LET TIME_INT=2.25; TIMER TIME_INT"

型変換

3458Aは、必要な場合は常に実数と整数間の変換を自動的に行います。実数が整数表現に変換されると、情報が失われる場合があります。この変換には2つの潜在的な問題があります。丸めエラーとレンジ・エラーです。

- 実数が整数に変換されると、実数値は最も近い整数に丸められます。小数点の右にある情報は失われます。
- レンジ・エラーは、実数値を整数値に変換するときに生じます。実数値の範囲はおおよそ-10^{308}～$+10^{308}$ですが、整数の範囲は–32768～+32767(おおよそ-10^4～$+10^4$)しかありません。したがって、すべての実数が対応する整数値に丸められる訳ではありません。この問題により、"Integer Overflow"エラーが生成される場合があります。

変数の使用

単純変数名と単純配列名の最大文字数は10です。最初の文字は英字(A～Z)でなければなりませんが、残りの9文字は英字、数字(0～9)、下線文字(_)または疑問符(?)のどれでもかまいません。大文字は小文字と同じです。**また変数名は、3458Aのコマンド、パラメータまたは保存済みステート名と同じであってはいけません。**

LETコマンド使用すれば任意の数値変数を割り当てることができます(キーワード"LET"が必要)。例えば、次の文は等価です。

```
OUTPUT 722; "LET TIME_INT = 120E-3"
OUTPUT 722; "LET TIME_INT =40*3E-3"
```

数値パラメータを使用する3458Aコマンドでは、数値パラメータを変数に置き換えることができます。(1)数値データの保存と(2)数値計算の2つ例があります。以下のセクションで、これら2つの使用法を説明します。

データ保存用変数

電源投入時は、3458Aにより生成される数値出力データはGPIB出力バッファに入れられ、そこからデータをシステム・コントローラに送ることができます。ただし、一部のアプリケーションでは、出力データをマルチメータの内部メモリに直接保存したい場合があります。ENTERコマンドは、読み取り値メモリから読み取り値を1つ取り出して(破壊的)、その値を指定した変数または配列位置へ入れます。

次のプログラムは、3458Aサブルーチン内でENTERコマンドを使用して読み取り値を保存します。

```
10 OUTPUT 722; "SUB DMM_CONF"
20 OUTPUT 722; "NRDGS 100"
30 OUTPUT 722; "TRIG SGL"
40 OUTPUT 722; "INTEGER I"
50 OUTPUT 722; "FOR I = 1 TO 100"
60 OUTPUT 722; " ENTER A[I]"
70 OUTPUT 722; "NEXT I"
80 OUTPUT 722; "SUBEND"
90 !
100 OUTPUT 722; "CALL DMM_CONF"
110 END
```

数値計算

変数は単純または配列かどうかに関係なく、数値計算で使用できます。3458Aコマンド・セットでは、データを操作するためにいくつかの演算関数が用意されています。3458Aの演算関数は、本章の後の部分で詳細に説明します。

マルチメータ値の読み取り

OUTPUTコマンドは、指定した変数の値を返します。下の例は、OUTPUTコマンドの使用方法を示します。

```
10 DIM A$[50]                       !コントローラ変数を宣言します。
20 OUTPUT 722; "LET VAL=COS(.5235)" !値を計算します。
30 OUTPUT 722; "OUTPUT VAL"         !結果を変数に読み込みます。
40 ENTER 722: A$                    !結果を入力します。
50 PRlNT A$                         !結果を出力します。
60 END
```

配列

3458Aのメモリ・スペースは、1次元配列用に割り当てることができます。実数配列の場合は、DIM *name(size)*コマンドまたはREAL *name(size)*コマンドを使用して配列を定義します。整数配列の場合は、INTEGER *name(size)*コマンドを使用します。すべての配列の添字の下限はゼロ(OPTION BASE 0)です。配列には、デフォルト・サイズはありません。例えば、10要素の配列を作成するには、下のようにサイズ9を指定します。

```
OUTPUT 722; "DIM TESTER(9)"
```

配列名には、数値変数名と同じルールが適用されます。ある配列要素を指定するには、括弧で囲んだ添字を指定する必要があります。添字の範囲は0～999の整数ですが、最大配列サイズは、3458Aの使用可能メモリ容量(ステートもサブプログラムも保存されてない場合は約10kバイト)で決まります。整数でない添字は、最も近い整

数に丸められます。

配列は、宣言し直すことによってサイズを変更できます。この場合、配列の個々の要素が値0に初期化されます。ただし、先にメモリをスクラッチしなければ、配列のタイプ(実数または整数)を定義し直すことはできません(第6章の「SCRATCHコマンド」を参照)。配列要素は、単純変数と同じように使用できます。

配列を埋める

配列要素は、宣言された場合(DIM、REALまたはINTEGERコマンド)またはサイズ変更された場合に、ゼロに初期化されます。配列を宣言したら、この配列に値をロードするにはFILL コマンドを使用します。FILLコマンドの構文は次のとおりです

FILL *array_name, list*

次のプログラムは、整数配列に整数値を埋めます。

```
10 OUTPUT 722; "INTEGER LIST(9)"
20 OUTPUT722; "FILL LIST 0,100,200,300,400,500,600,700,800,900"
30 END
```

注記

FILLコマンドは**注意して**使用してください。電源を入れ直すと、動作しません。この場合、コマンドは実質的にサブプログラムから削除されます。割り当てられた個々の値に対して、個別のLET文を使用してください。

配列のサイズ

SIZE?問合せコマンドは、指定した配列の要素数を返します。この数は、3458Aが使用するOPTION BASE 0変換のために、配列の最後の要素のインデックスより1大きくなります。したがって、10要素配列(例えばDIM LIST(9))を宣言した場合、SIZE?コマンドは"10"を返します。

次のプログラムは、要素が10個ある整数配列を定義して、SIZE?コマンドを使用して配列サイズを確認します。

```
10 OUTPUT 722; "INTEGER IARRAY(9)"
20 OUTPUT 722; "SIZE? IARRAY"
30 ENTER 722; A
40 PRINT A
50 END
```

配列と変数の消去

変数と配列はすべて3458Aの揮発性メモリに保存されます。3458Aの電源が切れると、変数と配列はすべて失われます。SCRATCHコマンドは、すべての変数、配列、サブプログラムおよび保存済みステート名(保存済みステートは第3章で説明しています)を消去します。

一般演算

一般演算式は、BASIC言語規約にしたがって、前面パネル・キーボード、システム・コントローラまたは3458Aサブプログラム内のいずれかから使用できます。標準演算子、一般演算関数、三角関数およびバイナリ関数が使用できます。3458Aには、簡単なカルキュレータ・モードもあります。

演算

標準演算子 (+ – * / ^)の他に、3458Aには2つの算術演算子があります。DIV(整数割り算)とMOD(モジュロ)です。単項の引き算は、次のように書きます。

```
A = 0-B
```

DIVコマンドは、割り算の整数部分を返します。通常の割り算が行われますが、小数点より右の桁はすべて(丸めではなく)切り捨てられます。次のプログラムは、7を3で割って、結果の整数部分(2)をシステム・コントローラに表示します。

```
10 OUTPUT 722; "OUTPUT(7 DIV 3)"
20 ENTER 722; A
30 PRINT "DIV Result ="; A
40 END
```

代表的な出力:

```
DIV Result = 2
```

MODコマンドは、割り算の余りを返します。DIVコマンドの場合と同様に、通常の割り算が行われますが、MODは余りだけを返します。次のプログラムは7を3で割って、結果の余り(1)をシステム・コントローラに表示します。

```
10 OUTPUT 722; "OUTPUT(7 MOD 3)"
20 ENTER 722; A
30 PRINT "MOD Result="; A
40 END
```

代表的な出力:

```
MOD Result=1
```

関係演算子(< > <= >= <>)と論理演算子(ANDおよびOR)も式の中で使用できます。

一般演算関数

次の表は、3458Aで使用できる一般演算関数です。引数("X"および"Y"で表します)は、数値、数値変数、関数、配列要素、または括弧で囲んだ数式のどれであってもかまいません。

関数/引数	意味
ABS(X)	引数の絶対値
SQR(X)	引数の正の平方根

対数関数

3458Aは、自然対数と常用対数の両方を計算できます。対数関数を次の表に示します。

関数/引数	意味
LOG(X)	Log_e(X): 正の引数のe(2.71828)を底とする自然対数

関数/引数	意味
EXP(X)	e^x: 逆対数。eを引数乗します。
LGT(X)	Log_{10}: 正の引数の10を底とする常用対数

三角関数

3458Aには、3つの三角関数があります。これらの三角関数を次の表に示します。

関数/引数	意味(Xはラジアン単位)
SIN(X)	引数のサイン
COS(X)	引数のコサイン
ATN(X)	引数のアークタンジェント

論理関数

3458Aには、4つの論理関数があります。AND(包含的AND)、OR(包含的OR)、EXOR(排他的OR)およびNOT(論理反転)です。最初の3つの関数は、2つの引数を比較して、それぞれの真理表に基づいて"0"または"1"を返します。引き数のゼロ以外の値(正または負)は、論理"1"とみなされます。ゼロだけが論理"0"として扱われます。

論理関数コマンドの構文は次のとおりです。4つの関数の真理表を下に示します。

argument **AND** *argument*

argument **OR** *argument*

argument **EXOR** *argument*

NOT argument

A	B	A AND B	A OR B	A EXOR B	NOT A	NOT B
0	0	0	0	0	1	1
0	1	0	1	1	1	0
1	0	0	1	1	0	1
1	1	1	1	0	0	0

バイナリ関数

3458Aには、7つのバイナリ関数があります。これらは、デジタル・パターンの生成に役立ちます。バイナリ関数を使用する場合、実数変数の引数値("X"および"Y")は–32768〜+32767の範囲の整数に丸められます。バイナリ関数を下の表に示します。

関数/引数	意味
BINAND(X,Y)	引数のビットごとの論理AND
BINCMP(X)	引数のビットごとのバイナリ補数
BINEOR(X,Y)	引数のビットごとの排他的論理OR
BINIOR(X.Y)	引数のビットごとの包含的論理OR

関数/引数	意味
BIT(X,ポジション)	引数の指定したビットの論理値を表す"0"または"1"を返します。ビット・ポジションは、0(Isb)〜15(msb)の範囲です。
ROTATE(X,変位)	**ビット回転を使用して、**指定した位置の数、引数を回転することにより得られる整数を返します。*
SHIFT(X,変位)	**ビット回転を使用しないで、**指定した位置の数、引数を回転することにより得られる整数を返します。*

* 変位が正の場合は、回転またはシフトは最下位ビットの方向です。
  変位が負の場合は、回転またはシフトは最上位ビットの方向です。

演算階層

3458Aは、括弧内の式を評価した後に、括弧外の演算関数を評価します。式の中に優先度が同じ演算子が2つ以上ある場合には、その階層は左から右です。

最高の優先度	括弧
	関数: SIN、COSなど
	累乗
	*、/、MOD、DIV
	+、–
最低の優先度	関連演算子: <、>、< =、>=など
	論理演算子: AND、ORなど

演算エラー

演算式を評価する場合、次のエラーが起こる可能性があります。3458Aは、演算エラーを他の実行エラーと全く同様に扱います。エラー処理の詳細については、第3章を参照してください。

エラーの名称
Division by Zero(ゼロによる割り算)
Real Overflow(実数オーバフロー)
Real Underflow(実数アンダフロー)
Integer Overflow(整数オーバフロー)
Square Root of a Negative Number(負の数値の平方根)
Log of a Non-Positive Number(正でない数値の対数)
Illegal Real Number(不正な実数値)
Trig Argument Out of Range(トリガ引数が範囲外)
BCD Exponent Too Big(BDC指数が大きすぎる)
HEX, Octal, or Decimal Argument Error(16進、8進または10進引数エラー)

比較をうまく行うには

整数値の算術比較を行う場合の特別な注意事項はありません。ただし、REAL数値、特に計算結果の数値を比較する場合には、丸めや他のシステムに固有の制限により、問題が生じる可能性があります。例えば、次の例のような、等しいかどうかを確認するIF...THEN文の使用を考えてみます。

```
10 OUTPUT 722; "SUB TESTER"
20 OUTPUT 722; "LET A=25.3765477"
30 OUTPUT 722; "IF SIN(A)^2 + COS(A)^2 = 1 THEN"
40 OUTPUT 722; " DISP 'EQUAL'"
50 OUTPUT 722; "ELSE"
60 OUTPUT 722; " DISP'NOT EQUAL'"
70 OUTPUT 722; " ENDIF"
80 OUTPUT 722; "SUBEND"
90 !
100 OUTPUT 722; "CALL TESTER"
110 END
```

丸め誤差、または有限マシンに固有の他の制限により生じる誤差のために、等しくないという結果になる場合があります。10進数の繰り返しや無理数は、3458Aのような有限マシンでは正しく表すことができません。

「等しい」のエラーの良い例は、数値の掛け算または割り算で起こります。整数でない値を掛けると、ほとんどと言って良いほど結果の小数点の右側の桁数は、掛け合わせた2つの数値のどちらもよりも大きくなります。

サブプログラム

3458Aは、BASIC言語サブプログラムを保存して実行することができます。このサブプログラムは、リモート・システム・コントローラ(HP シリーズ200/300コンピュータ)から3458Aメモリへダウンロードするか、前面パネル・キーボードから入力できます。このセクションでは、サブプログラムの構造と使用方法を説明します。また、サブプログラム内で使用するコマンドについても説明します。

サブプログラムは一連の3458Aコマンドで、SUBコマンドで始まり、SUBENDコマンドで終わります。SUBコマンドは名前をサブプログラムに割り当て、後でこのサブプログラムを実行する際にユーザがこの名前を使用します。サブプログラムは、3458Aの不揮発性メモリに保存されます。

3458Aにダウンロードされたサブプログラムは、システム・コントローラまたは前面パネル・キーボードから1つのコマンドで実行することができます。このため、3458Aが他のアクティビティでビジーの間でも、システム・コントローラが他のタスクを実行することが可能です。したがって、3458Aがそれ自体でタスクを実行する別のコンピュータのような役割を果たすため、システム・コントローラはマルチ・タスク機能を持つことになります。また、3458Aサブプログラム内のコマンドは、GPIBを使用して受信されたコマンドよりも速く実行されます。これは、3458Aがサブプログラム・コマンドを内部に保存する方法によるものです。

どのコマンドをサブプログラム内で使用できますか?

3458A用のほとんどのコマンドは、サブプログラム内に保存して実行することができます。保存できないコマンドは、CONTINUE、COMPRESS、DELSUBおよびSCRATCHだけです。3つの条件コマンドとループ・コマンドがサブプログラム内で使用するために用意されています。

異なるサブプログラムをいくつ保存できますか?

3458Aメモリに保存できるサブプログラムの正確な個数は、サブプログラムの個々のサイズに依存します。10個のコマンド(SUBコマンドとSUBENDコマンドを含む)が含まれる代表的なサブプログラムであれば、平均で約600バイトです。メモリ使用量の詳細については、第3章を参照してください。

サブプログラムをネストすることができますか?

はい。サブプログラムのネストとは、あるサブプログラムに別のサブプログラムを呼び出させる(実行させる)機能です。最大で10個のサブプログラムをネストできます。

サブプログラムの作成とロード

このセクションのサブプログラムのサンプル・プログラムは、比較的単純な3458A演算です。これをコピーして、独自デザインのより複雑なメインライン・プログラムの中で使用することができます。このセクションでは、サブプログラムの作成と編集の方法も説明します。

注記 プログラムミングのヒント
テスト・システム・プログラムの始めるにあたっては、SCRATCHコマンドを実行して、サブプログラムをシステム・コントローラからダウンロードする必要があります。これにより、3458Aのメモリ管理が行われ、サブプログラムが確実にダウンロードされて、必要な時に使用できる状態になります。

SUBコマンドを実行すると、サブプログラム内の、SUBENDコマンドまでのすべての後続コマンドを保存するように、3458Aに対して命令がなされます。

サブプログラム名の最大文字数は10です。最初の文字は英字(A〜Z)でなければなりませんが、残りの9文字は英字、数字(0〜9)、下線文字(_)または疑問符(?)のどれでもかまいません。また変数名が、3458Aのコマンドあるいはパラメータ、定義済みの配列名あるいは変数名、または保存済みステート名と同じであってはいけません。

次のプログラムは、3つのDC電圧測定を行うようにマルチメータを構成するサブプログラムの作成方法を示します。

```
10 OUTPUT 722; "SUB DMM_CONF"
20 OUTPUT 722; "DCV8,0.00125"
30 OUTPUT 722; "NRDGS 3"
40 OUTPUT 722; "TRIG SGL"
50 OUTPUT 722; "SUBEND"
60 END
```

2つの文、SUB DMM_CONFおよびSUBENDと、行20、行30および行40の3つのコマンドが、DMM_CONFという名前のサブプログラムを形成しています。

サブプログラムが入力されると、3458Aが他のコマンドと同様に構文エラーがないか確認します。構文が正しくない場合には、エラーが生成され、そのコマンドはサブプログラムに保存されません。この場合、サブプログラムをシステム・コントローラで編集して、再度ダウンロードする必要があります。3458Aは、サブプログラムを不揮発性メモリに保存します。さらに、前面パネル・キーボードまたはシス

テム・コントローラのいずれかからサブプログラムを実行できます。SUBENDコマンドの実行時にサブプログラム・ネスト・エラーが存在すれば(呼び出されたサブプログラムが3458Aメモリに存在しないなど)、サブプログラムは保存されません。

3458Aメモリにすでに存在するサブプログラム名を使用してサブプログラムを作成またはダウンロードすると、新しいサブプログラムが前のサブプログラムを上書きします。

サブプログラム・コマンドのタイプ

3458Aのサブプログラム関連コマンドは、サブプログラム内でのみ使用されます。サブプログラムの定義コマンドと削除コマンドは、保存、表示、サブプログラムの内部メモリからの削除を処理します。実行コマンドは、サブプログラムの内部または外部からサブプログラムの実行を制御します。

定義/削除コマンド

サブプログラムの定義コマンドと削除コマンドは、サブプログラムの始まりと終わりを識別し、サブプログラムのメモリへの保存とメモリからの削除を行います。また、内部メモリに現在保存されているサブプログラムをリストします。

サブプログラムの定義コマンドと削除コマンドの構文は次のとおりです。

SUB *sub_name*

SUBEND

DELSUB *sub_name*

SCRATCH

CAT

LIST *sub_name*

COMPRESS *sub_name*

SUB/SUBEND

すべての3458AサブプログラムにSUBコマンドとSUBENDコマンドが含まれていなければなりません。またSUBコマンドは、すべての3458Aサブプログラムの最初の行になければなりません。これがサブプログラムがどこで始まるのかを識別し、名前をサブプログラムに割り当てます。SUBコマンドが実行されると、3458Aがサブプログラムの内部メモリへの保存を開始します。

SUBENDコマンドは、すべての3458Aサブプログラムの最後の行になければなりません。これが、サブプログラムがどこで終わるのかを識別し、またサブプログラムのエントリをターミネートします。サブプログラムが実行されるたびに、SUBコマンドとSUBENDコマンドの間にリストされているコマンドが順に実行されます。

SUBコマンドとSUBENDコマンドは、1つのサブプログラムの中でそれぞれ1つずつしか使用できません。この他にSUBコマンドとSUBENDコマンドがあるとエラーが生成されます。

DELSUB

DELSUB(サブプログラム削除)コマンドは、指定したサブプログラムを内部メモリから削除しますが、サブプログラム名自体はサブプログラムのカタログ・リスト

(CATコマンド)から削除しません。

SCRATCH SCRATCHコマンドは、すべての3458Aサブプログラム、変数および 配列を内部メモリから削除(スクラッチ)します。また、すべての名前定義をカタログ・リスト(CATコマンド)から削除します。サブプログラムの実行中にSCRATCHが実行されると、エラーが生成されますが、サブプログラムはメモリから消去されません。

CAT CAT(カタログ)コマンドは、すべての3458Aサブプログラムの名前、単純変数、保存済みステートおよび内部メモリ内に現在保存されている配列をリストします。リストする配列またはサブプログラムがなくなった場合、CATコマンドがワード"DONE"を返します。保存済みステートの詳細については、第3章を参照してください。カタログのフォーマットは次のとおりです。

サブプログラム*:* **SUB** *sub_name*

整数配列*:* **IARRAY** *array_name*

実数配列*:* **RARRAY** *array_name*

保存済みステート*:* **STATE** *state_name* (不揮発性メモリ)

単純変数*:* **INT** *variable_name*
REAL *variable_name*

次のプログラムは、CATコマンドの使用方法を示します。

```
10 DIM A$[80]
20 OUTPUT 722; "CAT"
30 REPEAT
40   ENTER 722; A$
50   PRINT A$
60 UNTIL A$="DONE"
70 END
```

LIST LISTコマンドを使用すると、指定したサブプログラムをリストできます。サブプログラムは前面パネルから編集できないことを念頭においてください。システム・コントローラから編集する必要があります。次のプログラムは、サブプログラムDMM_CONFをシステム・コントローラに対してリストする方法を示します。

```
10 DIM A$[100]
20 OUTPUT 722; "LIST DMM_CONF"
30 REPEAT
40   ENTER 722; A$
50   PRINT A$
60 UNTIL A$="SUBEND"
70 END
```

COMPRESS COMPRESSコマンドは、指定したサブプログラムのテキストを内部メモリから削除します(サブプログラムは不揮発性メモリには保存されなくなり、電源が切断されると失われます)。これにより内部メモリのスペースが節約されますが、サブプログラムをリストする機能(LISTコマンド)はなくなります。COMPRESSコマンドは、サブプログラムのデバッグとテストの終了後にのみ使用してください。

実行コマンド

サブプログラム実行コマンドは、サブプログラムの実行を制御します。サブプログラム実行コマンドの構文は次のとおりです。

CALL *sub_name*

PAUSE

CONT

サブプログラムCALL

CALLコマンドは、名前付きサブプログラムを実行して、完了を待った後、他のコマンドを実行します。これは、サブプログラムが終了するまで、後続のコマンドは(GPIBインタフェースまたは前面パネル・キーボードのいずれからも)受け付けられないということです。レディ・ビット(3458Aステータス・レジスタのビット4)は、サブプログラムの実行中は"0"のままとなります。サブプログラムが実行を終了すると、レディ・ビットは"1"に設定されて、3458Aが別のコマンドを受信できる状態にあることを示します。

CALLコマンドは、サブプログラムの中で別のサブプログラムを呼び出すためにも使用できます。これによって、"ネストされた"サブプログラムの機能を拡張することができます。ネストされたサブプログラムを使用する場合は、一度に1つのサブプログラムだけが実行されるように、呼び出し側のサブプログラムは中断されます。サブプログラムは、深さ10までネストできます。

サブプログラムPAUSE

PAUSEコマンドは、CALLコマンドによって最後に実行されたサブプログラムを休止します。サブプログラムが一時停止されると、実行を再開するにはCONT(継続)コマンドを実行する必要があります。CONTコマンドを使用すれば、サブプログラムが、PAUSEコマンドの後の、次のコマンドから始まって、完了まで実行を続行することができます。

サブプログラムが一時停止されていないときにCONTコマンドを実行しようとすると、エラーが生成されます。

サブプログラムの一時停止／実効状態を知る

PAUSED?問合せコマンドは、サブプログラムが現在一時停止されている場合には"1"を返し、サブプログラムが実行されている場合(または、実行が終了した場合)には"0"を返します。次のプログラムは、PAUSED?コマンドの使用方法を示します。

```
10 OUTPUT 722; "RUN DMM_CONF;PAUSE"
20 OUTPUT 722; "PAUSED?"
30 ENTER 722; A
40 IF A=1 THEN PRINT "SUBPROGRAM IS PAUSED"
50 IF A=0 THEN PRINT "SUBPROGRAM IS NOT PAUSED"
60 END
```

サブプログラムの中止

GPIB CLEARコマンド(付録Bを参照)は、CALLコマンドを使用して実行したサブプログラムの実行を中止します。これによって、制御がGPIBコマンド入力バッファまたは前面パネル・キーボードに戻ります。

サブプログラムを抜ける

サブプログラムは、SUBENDコマンドに達するまで実行を継続します。その後、そのサブプログラムを呼び出したサブプログラム(ネストされたサブプログラム)、またはGPIB入力バッファあるいは前面パネル・キーボード(サブプログラムを実行した方)に制御が戻ります。RETURNコマンドも、サブプログラムの終了に使用でき

ます。例えば、サブプログラムの条件付き終了が必要な場合は、サブプログラム内のIF...THENループの中にRETURNを置きます。RETURNコマンドは、SUBENDコマンドを実行しないで制御を呼び出し側に戻します。例えば、

```
10 OUTPUT 722; "SUB DMM_CONF"
20 OUTPUT 722; "DCV 8, 0.00125"
30 OUTPUT 722; "TRIG SGL"
40 OUTPUT 722; "ENTER A'
60 OUTPUT 722; "IF A<5.06 THEN; RETURN"
70 OUTPUT 722; "ELSE"
80 OUTPUT 722; "TRIG SGL"
90 OUTPUT 722; "ENDIF"
100 OUTPUT 722; "SUBEND"
110 !
120 OUTPUT 722; "CALL DMM_CONF"
130 END
```

サブプログラムのネスト

あるサブプログラムが、その実行が終了する前に別の(ネストされた)プログラムを呼び出すことが可能です。この別のサブプログラムがSUBENDコマンドを実行すると、元のサブプログラムに埋め込まれたCALLコマンドの次のコマンドから実行を継続します。

3458Aには、サブプログラムのネストに2つの要件があります。1番目は、別のサブプログラム内から呼び出されるサブプログラムは、呼び出しを行うサブプログラムを保存する前に内部メモリに保存しなければならないことです。これは、3458Aがサブプログラムを保存すると同時に各コマンドの構文をチェックするためです。埋め込まれたCALLコマンドが現れると、3458Aはサブプログラムがその名前でメモリ内に存在するかどうかをチェックします。存在しなければ、エラーが生成されます。2番目は、サブプログラムは10レベルを超える深さではネストできないことです。あるサブプログラムを別のサブプログラム内に入れることはできません。例えば、次のプログラムはエラーを生成します。

```
10 OUTPUT 722; "SUB DMM_CONF"
20 OUTPUT 722;"DCV8,0.00125"
30 OUTPUT 722; "SUB TESTER"  !この結果はエラーになります。
40 OUTPUT 722; "SUBEND"
50 !
60 OUTPUT 722; "CALL DMM_CONF"
70 END
```

サブプログラム内の条件文

3458Aでは、条件付き分岐とループに3つのBASIC言語文が用意されています。これらの文は3458Aサブプログラム内でのみ使用してください。条件付き分岐文とループ文は、繰り返しテスト、配列の初期化などに使用します。

3つの条件文とは、FOR...NEXT、WHILE...ENDWHILEおよびIF...THENです。これらの文は拡張BASIC言語で使用されているものと似ています。唯一の例外は、3458Aサブプログラムには、行番号と分岐用のGOTO分がないことです。ループ文と条件付き分岐文は、深さ7までネストできます。

FOR...NEXTループ

FOR...NEXTコマンドは、ループ・カウンタが指定値を超えるまで繰り返されるルー

プを定義します。FOR...NEXT コマンドの構文は次のとおりです。

FOR *counter* = *initial_value* **TO** *final_value* [**STEP** *step_size]*

プログラム・セグメント

NEXT *counter*

*counter*パラメータは、ループ・カウンタの役割をする変数です。*initial_value*パラメータと*final_value*パラメータは、数値、数値変数または数値式のいずれでもかまいません。オプションの*step_size*パラメータは、数値、またはループをパスするごとのループ・カウンタの増分を指定する数値式のいずれでもかまいません。*step_size*が負の値の場合は、ループ・カウンタが減ります。プログラム・セグメントは、ループ・カウンタが*final_value*を超えるまで繰り返し実行されます。

```
10 OUTPUT 722; "SUB DMM_CONF"
20 OUTPUT 722; "NRDGS 100"
30 OUTPUT 722; "TRIG SGL"
40 OUTPUT 722; "INTEGER I"
50 OUTPUT 722; "FOR I = 1 TO 100"
60 OUTPUT 722; " ENTER A[ I]"
70 OUTPUT 722; "NEXT I"
80 OUTPUT 722; "SUBEND"
90 !
100 OUTPUT 722; "CALL DMM_CONF"
110 END
```

WHILEループ

WHILEコマンドは、指定した数値式がtrueの場合にループを定義します。WHILEコマンドの構文は次のとおりです。

WHILE *expression*

プログラム・セグメント

ENDWHILE

WHILE演算は、ループの始めに実行されるテストの結果に依存します。テストが真(ゼロに等しくない)の場合、WHILE文とENDWHILE文の間のプログラム・セグメントが実行され、分岐がWHILE文に戻ります。テストが偽(ゼロに等しい)の場合、ENDWHILE文に続く文でプログラムの実行が継続されます。

```
 10 OUTPUT 722; "SUB DMM_CONF"
 20 OUTPUT 722; "INTEGER I"
 30 OUTPUT 722; "LET I=1"
 40 OUTPUT 722; "NRDGS 100"
 50 OUTPUT 722; "TRIG SGL"
 60 OUTPUT 722; "WHILE I <=100"
 70 OUTPUT 722; " ENTER A[I]"
 80 OUTPUT 722; " LET I=I+1"
 90 OUTPUT 722; "ENDWHILE"
100 OUTPUT 722; "SUBEND"
110 !
120 OUTPUT 722; "CALL DMM_CONF"
130 END
```

IF...THEN分岐

IF...THENコマンドは、3458Aサブプログラム内の条件分岐を実現します。IF...THENコマンドの構文は次のとおりです。

IF *expression* **THEN**

プログラム・セグメント

[**ELSE**]

[program segment]

ENDIF

ENDIF文が、サブプログラム内のIF...THEN文の次になければばなりません。ELSEはオプション文ですが、使用する場合にはENDIF文の前になければばなりません。式の結果がtrue(ゼロに等しくない)の場合には、IF...THEN文の後から、ELSE文とENDIF文の前にあるすべてのコマンドが実行されます。

式がtrueの場合、IF...THENとELSEの間のプログラム・セグメントで実行が継続されます。式がfalseの場合、ELSEの後のセグメントで実行が継続されます。いずれの場合も、プログラム・セグメントが完了すると、他のループと条件付き分岐がないと仮定して、ENDIF文に続く文でプログラム実行が継続されます。

```
10 OUTPUT 722; "SUB DMM_CONF"
20 OUTPUT 722; "INTEGER I"
30 OUTPUT 722; "LET I=1"
40 OUTPUT 722; "NRDGS 100"
50 OUTPUT 722; "TRIG SGL"
60 OUTPUT 722; "IF I<100 THEN"
70 OUTPUT 722; " ENTER A[I] "
80 OUTPUT 722; " LET l=l+1"
90 OUTPUT 722; "ENDIF"
100 OUTPUT 722; "SUBEND"
110 !
120 OUTPUT 722; "CALL DMM_CONF"
130 END
```

付録A　仕様

概要

直流電圧／電流、抵抗測定における確度は、読み取り値のppm＋レンジのppmで、また交流電圧／電流測定における確度は、読み取り値の%＋レンジの%で表示されています。レンジは1V、10Vなどのスケール名を意味しており、1.2V、12Vなどのフル・スケールを意味するものではありません。確度は最近の校正から特定時間後について仕様化されています。

絶対確度と相対確度

3458Aのすべての確度は、校正標準器に対する相対確度として仕様化されています。この相対確度に校正標準器のトレーサビリティを加えることにより、3458Aの絶対確度が求められます。例えば直流電圧測定では、Agilent工場で2ppmのトレーサビリティを持っています。つまり、NBS(米国標準局)に対する3458Aの絶対確度は、直流電圧測定の相対確度にこの2ppmを加えたものとなります。3458Aを校正する場合、そのトレーサビリティは使用する校正標準器のトレーサビリティに依存します。このトレーサビリティはAgilentの持つ2ppmと異なる場合もあります。

例1: 24時間の相対確度; 動作温度Tcal±1ºCの場合

測定時の周囲温度が校正温度(Tcal)の±1ºC以内、10Vレンジで10Vの直流電圧を測定した場合、24時間の相対確度は0.5ppm＋0.05ppmです。つまり、

読み取り値の0.5ppm + レンジの0.05ppm

測定時に生じる誤差は、相対確度では、

(0.5/1,000,000×10V) + (0.05/1,000,000×10V)
= ±5.5µVまたは10Vの0.55ppm

温度変化による誤差

3458Aの仕様は、自動校正(ACAL)後24時間以内、および±1ºC以内の周囲温度変化に基づきます。3458AのACAL機能は、時間や温度の変化により生じる測定誤差を校正します。

以下の例は、異なった温度状況における3458Aの測定誤差を計算することにより、自動校正の誤差補正を示しています。各例では以下の一定条件を使用しています。

10V直流電圧入力
10V直流電圧レンジ
Tcal ＝ 23ºC
90日間の確度

例2: 動作温度28ºC:

ACAL使用の場合

以下は、28ºCの動作温度、ACAL使用時の3458Aの相対確度です。結果は、2桁に丸めらています。

(4.1ppm×10V) + (0.05ppm×10V) = 42µV

全体の相対誤差 = 42µV

例3: 動作温度38ºC:

ACALなしの場合

動作温度が38ºCの場合、Tcal±1ºCより14ºC大きい場合、ACALなしの温度係数による追加誤差が加算されます。

(4.1ppm×10V) + (0.05ppm×10V) = 42µV

温度係数(/ ºC)による誤差

(0.5ppm×10V+0.01ppm×10V)×14ºC = 71µV

全体の相対誤差 = 113µV

例4: 動作温度38ºC:

ACAL使用の場合

例4は、ACAL使用を除いて例3と同じ条件です。ACAL使用により、温度により生じる誤差を減少させることができます。動作温度はTcal±5ºCの標準範囲を10ºC超えています。

(4.1ppm×10V) + (0.05ppm×10V) = 42µV

温度係数(/ ºC)による誤差:

(0.15ppm×10V+0.01ppm×10V)×10ºC =16µV

全体の相対誤差 = 58µV

例5: 90日間の絶対確度;

例5では、例4と同じ条件下で、トレーサビリティを加えることにより絶対確度を求めています。

(4.1ppm×10V) + (0.05ppm×10V) = 42µV

温度係数(/ ºC)による誤差:

(0.15ppm×10V+0.01ppm×10V)×10ºC =16µV

Agilent工場のもつ読み取り値の2ppmのトレーサビリティ：

(2ppm×10V) = 20µV

全体の相対誤差 = 78µV

追加誤差

3458Aを100未満の電源ライン周波数で動作させると、ノイズと利得が有意になるため追加誤差が加わります。例6は、0.1 PLCでの誤差補正を示します。

例6: 動作温度は28ºC、0.1 PLC:

例2と同じ条件を想定しますが、この場合は追加誤差を追加します。

(4.1ppm×10V) t (0.05ppm×10V) = 42µV

追加誤差チャートとRMSノイズ係数表を参照すると、0.1 PLCでの追加誤差は次のようになります。

(2ppm×10V) + (0.4ppm×1×3×10V) = 32µV

全体の相対誤差 = 74µV

1 / 直流電圧測定

直流電圧測定

レンジ	フル・スケール	最大分解能	入力抵抗	温度係数 (読み取り値のppm＋レンジのppm)/ ºC	
				ACALなし[1]	ACALあり[2]
100mV	120.00000	10nV	＞10GΩ	1.2 + 1	0.15 + 1
1V	1.20000000	10nV	＞10GΩ	1.2 + 0.1	0.15 + 0.1
10V	12.0000000	100nV	＞10GΩ	0.5 + 0.01	0.15 + 0.01
100V	120.000000	1μV	10MΩ± 1%	2 + 0.4	0.15 + 0.1
1000V	1050.00000	10μV	10MΩ± 1%	2 + 0.04	0.15 + 0.01

確度 [3] (読み取り値のppm (オプション002の読み取り値のppm) + レンジのppm)

レンジ	24時間 [4]	90日間 [5]	1年間 [5]	2年間 [5]
100mV	2.5 + 3	5.0 (3.5)+ 3	9 (5)+ 3	14 (10)+ 3
1V	1.5 + 0.3	4.6 (3.1)+0.3	8 (4)+ 0.3	14 (10)+0.3
10V	0.5 + 0.05	4.1 (2.6) + 0.05	8 (4) + 0.05	14 (10)+0.05
100V	2.5 + 0.3	6.0 (4.5) + 0.3	10 (6)+0.3	14 (10)+ 0.3
1000V [6]	2.5 + 0.1	6.0 (4.5)+ 0.1	10 (6)+ 0.1	14 (10)+ 0.1

トランスファ確度／直線性

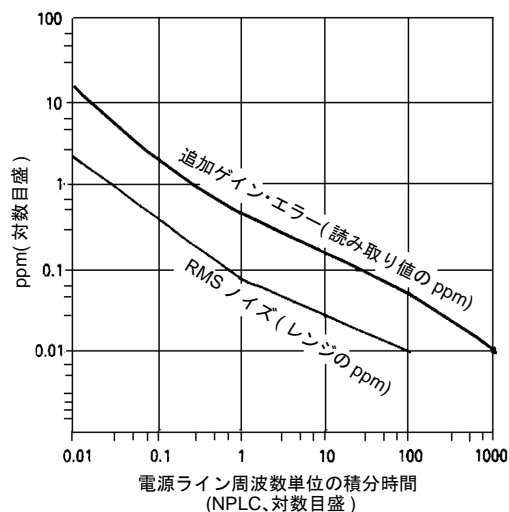
レンジ	10分間、Tref±0.5ºC (読み取り値のppm ＋レンジのppm)
100mV	0.5 + 0.5
1V	0.3 + 0.1
10V	0.05 + 0.05
100V	0.5 + 0.1
1000V	1.5+0.05

状態

- 4時間のフォーム・アップ後、フル・スケールからフル・スケールの0.1倍まで。
- 1000Vレンジでの測定は、最初の測定値の5%以内で、かつ次に続く測定値が安定することが必要。
- Trefは測定開始時の環境温度。
- 測定は、認められている計測法を使用して固定レンジ(＞4min)で行う。

セトリング特性

最初の読み取り値あるいはレンジ変更の誤差は、入力電圧の0.0001%の追加誤差を加算。
読み取りのセトリング時間は、信号源インピーダンスおよびケーブルの誘電吸収の特性に影響を受ける。

追加誤差

ノイズ除去比(dB) [7]

	AC NMR [8]	AC ECMR	DC ECMR
NPLC ＜ 1	0	90	140
NPLC ＞ 1	60	150	140
NPLC ＞ 10	60	150	140
NPLC ＞ 100	60	160	140
NPLC = 1000	75	170	140

*RMSノイズ

レンジ	乗数
0.1V	×20
1V	×2
10V	×1
100V	×2
1000V	×1

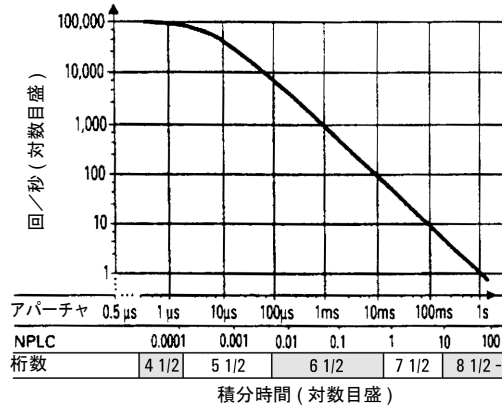
RMSノイズ誤差の場合は、グラフのRMSノイズの結果にチャートの係数を掛けます。ピーク・ノイズ誤差の場合は、RMSノイズに3を掛けます。

1. ACALなしの場合はTcal ±1ºCからの追加誤差。
2. ACALありの場合はTcal ±5ºCからの追加誤差。
3. PRESET、NPLC100実行時の仕様。
4. 固定レンジ(＞4分)、MATH NULL、Tcal±1ºCの場合
5. 90日間、1年間、2年間の確度は、ACAL後24時間以内かつ± 1ºC、Tcal ± 5ºC、MATH NULLおよび固定レンジの場合

   オプション002高安定基準では、90日間の確度から読み取り値の1.5ppmを、1年間または2年間の確度から読み取り値の4ppmを減ずる。

   MATH NULLなしの場合、0.15レンジのppmを10Vに、0.7レンジのppmを1Vに、7レンジのppm を0.1Vに追加します。MATH NULLありで、4分未満の固定レンジの場合、レンジの0.25ppmを10Vに、レンジの1.7ppmを1V、レンジの17ppm を0.1Vに追加します。

   NBS(米国標準局)に対するAgilent工場のトレーサビリティは、読み取り値の2ppmの追加誤差を加算。トレーサビリティは国家標準に対する絶対誤差で、外部校正用装置に関連する
6. 100Vを超える入力では、10ppm×$(Vin/1000)^2$を加算。
7. Lo端子に1kΩ を接続した不平衡入力時。LFREQに電源周波数の±0.1%を設定。
8. 電源周波数±1%の場合、AC NMR はNPLC ≧ 1 で40dB、NPLC≧100で80dB。電源周波数±5%の場合、NPLC≧100で65dB。

読み取り速度(AUTO ZEROオフ)

温度係数(AUTO ZEROオフ)

±1ºCの安定環境下の場合、AUTO ZEROオフでは以下の追加誤差を加算。

レンジ	追加誤差
100mV〜10V	5µV/ ºC
100V〜1000V	500µV/ ºC

読み取り速度[1]

NPLC	アパーチャ	桁数	ビット	回／秒 A-Zero オフ	回／秒 A-Zero オン
0.0001	1.4µs	4.5	16	100,000[3]	4,130
0.0006	10µs	5.5	18	50,000	3,150
0.01	167µs[2]	6.5	21	5,300	930
0.1	1.67ms[2]	6.5	21	592	245
1	16.6ms[2]	7.5	25	60	29.4
10	0.166s[2]	8.5	28	6	3
100		8.5	28	36/min	18/min
1000		8.5	28	3.6/min	1.8/min

1. PRESET, DELAY 0, DISP OFF, OFORMAT DINTの場合。
2. アパーチャ時間はLFREQに独立して設定可能。これらのアパーチャ時間は60Hzに対する時間。NPLCの値は1 NPLC=1/LFREQ。50Hzでは､アパーチャ時間を1.2倍、読み取り速度を0.833倍する。
3. OFORMAT SINTの場合。

最大入力

	定格入力	非破壊入力
HI-LO間	±1000Vpk	±1200Vpk
LO-Guard間[4]	±200Vpk	±350Vpk
Guard-Earth間[5]	±500Vpk	±1000Vpk

4. ＞10^{10}Ω LO-アース、ガード・オープン時
5. ＞10^{12}Ωガード-アース

入力端子

端子材質: 金めっきテルリウム銅

入力リーク電流: 20pA、25 ºC

2 / 抵抗測定

2端子、4端子(OHM、OHMF機能)

レンジ	フル・スケール	分解能	測定電流[6]	テスト電圧	開放時の端子間電圧	最大リード抵抗 (OHMF)	最大直列オフセット (OCOMP ON)	温度係数 (読み取り値のppm＋レンジのppm)/ ºC ACALなし[7]	温度係数 (読み取り値のppm＋レンジのppm)/ ºC ACALなし[8]
10Ω	12.00000	10µΩ	10mA	0.1V	12V	20Ω	0.01V	3+1	1+1
100Ω	120.00000	10µΩ	1mA	0.1V	12V	200Ω	0.01V	3+1	1+1
1kΩ	1.2000000	100µΩ	1mA	1.0V	12V	150Ω	0.1V	3+0.1	1+0.1
10kΩ	12.000000	1mΩ	100µA	1.0V	12V	1.5kΩ	0.1V	3+0.1	1+0.1
100kΩ	120.00000	10mΩ	50µA	5.0V	12V	1.5kΩ	0.5V	3+0.1	1+0.1
1MΩ	1.2000000	100mΩ	5µA	5.0V	12V	1.5kΩ		3+1	1+1
10MΩ	12.000000	1Ω	500nA	5.0V	12V	1.5kΩ		20+20	5+2
100MΩ[9]	120.00000	10Ω	500nA	5.0V	5V	1.5kΩ		100+20	25+2
1GΩ[7]	1.2000000	100Ω	500nA	5.0V	5V	1.5kΩ		1000+20	250+2

6. 測定電流は±3%の絶対確度。
7. ACALなしの場合はTcal±1ºCからの追加誤差。
8. Tcal±5ºCからの追加誤差。
9. 被測定抵抗に並列10MΩの抵抗が接続される。

2 確度 [1] (読み取り値のppm＋レンジのppm)

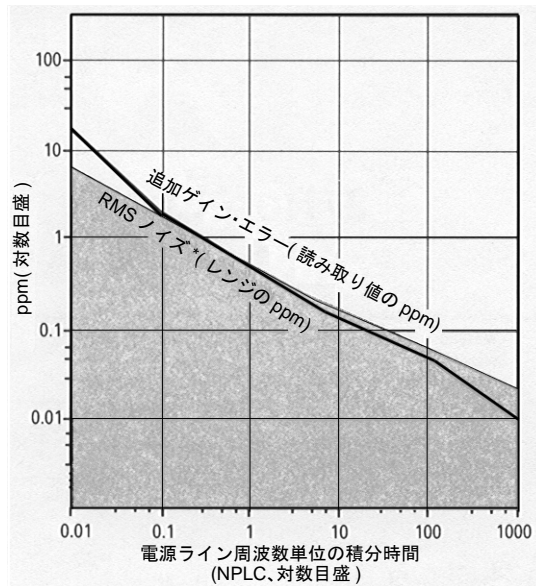
レンジ	24時間 [2]	90日間 [3]	1年間 [3]	2年間 [3]
10Ω	5+3	15+5	15+5	20+10
100Ω	3+3	10+5	12+5	20+10
1kΩ	2+0.2	8+0.5	10+0.5	15+1
10kΩ	2+0.2	8+0.5	10+0.5	15+1
100kΩ	2+0.2	8+0.5	10+0.5	15+1
1MΩ	10+1	12+2	15+2	20+4
10MΩ	50+5	50+10	50+10	75+10
100MΩ	500+10	500+10	500+10	0.1%+10
1GΩ	0.5%+10	0.5%+10	0.5%+10	1%+10

1. PRESET, NPLC 100, OCOMP ON実行時の仕様。
2. Tcal ±1ºC
3. 90日間、1年間、2年間の確度は、ACAL後24時間以内かつ±1ºC、Tcal±5ºC。

   NBS(米国標準局)に対する10kΩのAgilent工場のトレーサビリティは、読み取り値の3ppmの追加誤差を加算。トレーサビリティは国家標準に対する絶対誤差で、校正用装置に関連する。

2端子抵抗確度

2端子抵抗(OHM)確度は、4端子抵抗(OHMF)確度に以下のオフセット誤差を追加。
24時間: 50mΩ、90日間: 150mΩ、1年間: 250mΩ、2年間: 500mΩ

追加誤差

*RMSノイズ	
レンジ	乗数
10Ω、100Ω	×10
1kΩ～100kΩ	×1
1MΩ	×1.5
10MΩ	×2
100MΩ	×120
1GΩ	×1200

RMSノイズ誤差の場合は、グラフのRMSノイズの結果にチャートの係数を掛けます。ピーク・ノイズ誤差の場合は、RMSノイズに3を掛けます。

セトリング特性

レンジ変更後の最初の測定値の誤差は、電流レンジの90日間の測定誤差を加算。前もってプログラムされたセトリング遅延時間は、＜200pFの外部回路の容量にあてはまる。

読み取り速度 [4]

			回／秒	
NPLC [5]	アパーチャ	桁数	Auto-Zero オフ	Auto-Zero オン
0.0001	1.4µs	4.5	100,000 [7]	4,130
0.0006	10µs	5.5	50,000	3,150
0.01	167µs [6]	6.5	5,300	930
0.1	1.66ms [6]	6.5	592	245
1	16.6ms [6]	7.5	60	29.4
10	0.166s [6]	7.5	6	3
100		7.5	36 /min	18/min

4. PRESET, DELAY 0, DISP OFF, OFORMAT DINTの場合。

   OHMF, OCOMP ONでの読み取り速度は遅くなる。
5. NPLC＜1での抵抗測定は、まわりのノイズ環境の影響を受ける。その為、測定確度を維持させるには、適切なシールディングやガーディングを施す必要あり。
6. アパーチャ時間はLFREQに独立して設定可能。これらのアパーチャ時間は60Hzに対する時間。NPLCの値は1 NPLC＝1/LFREQ。50Hzでは、アパーチャ時間を1.2倍、読み取り速度を0.833倍する。
7. OFORMAT SINTの場合。

   *テフロンは、E. I. duPont deNemours and Coの登録商標です。

測定における注意点

テフロン*ケーブル、またはこれらの測定用の他のハイ・インピーダンス、低誘電吸収ケーブルを使用することを推奨します。

最大入力

	定格入力	非破壊入力
HI-LO間	±1000Vpk	±1000Vpk
HI, LO Sense-LO間	±200Vpk	±350Vpk
LO-Guard間	±200Vpk	±350Vpk
Guard-Earth間	±500Vpk	±1000Vpk

温度係数 (Auto-Zeroオフ)

±1ºCの安定環境下の場合、AUTO ZEROオフでは以下の追加誤差(レンジのppm)/ ºCを加算。

レンジ	誤差
10Ω	50
100Ω	50
1kΩ	5
10kΩ	5
100kΩ	1

レンジ	誤差
1MΩ	1
10MΩ	1
100MΩ	10
1GΩ	100

3 / 直流電流測定

直流電流測定 (DCIファンクション)

レンジ	フル・スケール	分解能	シャント抵抗	負荷電圧	温度係数 (読み取り値のppm + レンジのppm)/ ºC	
					ACALなし[1]	ACALあり[2]
100nA	120.000	1pA	545.2kΩ	0.055V	10+200	2+50
1µA	1.200000	1pA	45.2kΩ	0.045V	2+20	2+5
10µA	12.000000	1pA	5.2kΩ	0.055V	10+4	2+1
100µA	120.00000	10pA	730Ω	0.075V	10+3	2+1
1mA	1.2000000	100pA	100Ω	0.100V	10+2	2+1
10mA	12.000000	1nA	10Ω	0.100V	10+2	2+1
100mA	120.00000	10nA	1Ω	0.250V	25+2	2+1
1A	1.0500000	100nA	0.1Ω	＜1.5V	25+3	2+2

確度[3] (読み取り値のppm + レンジのppm)

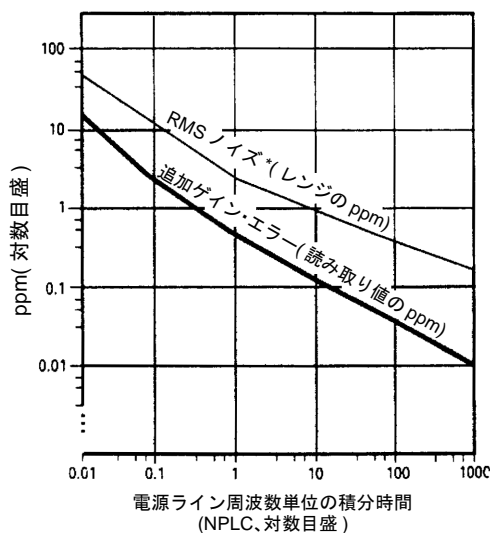
レンジ	24時間[4]	90日間[5]	1年間[5]	2年間[5]
100nA[6]	10+400	30+400	30+400	35+400
1µA[6]	10+40	15+40	20+40	25+40
10µA[6]	10+7	15+10	20+10	25+10
100µA	10+6	15+8	20+8	25+8
1mA	10+4	15+5	20+5	25+5
10mA	10+4	15+5	20+5	25+5
100mA	25+4	30+5	35+5	40+5
1A	100+10	100+10	110+10	115+10

1. ACALなしの場合はTcal ±1ºCからの追加誤差。
2. ACALありの場合はTcal ±5ºCからの追加誤差。
3. PRESET, NPLC 100実行時の仕様。
4. Tcal±1ºC。
5. 90日間、1年間、2年間の確度は、ACAL後24時間以内かつ±1ºC、Tcal±5ºC。

   NBC(米国標準局)に対するAgilent工場のトレーサビリティは、読み取り値の5ppmの追加誤差を加算。トレーサビリティは10Vと10kΩのトレーサビリティの合計。
6. 代表値。

セトリング特性

最初の測定値あるいはレンジ変更の誤差は、入力電流の0.001%の追加誤差を加算。読み取り値のセトリング時間は、信号源インピーダンスやケーブルの誘電吸収の特性に影響を受ける。

追加誤差

*RMSノイズ	
レンジ	乗数
100nA	×100
1µA	×10
10µA～1A	×1

RMSノイズ誤差の場合、グラフのRMSノイズの結果にチャートの係数を掛けます。ピーク・ノイズ誤差の場合、RMSノイズ誤差に3を掛けます。

測定における注意点

低電流測定には、ハイ・インピーダンス、低誘電吸収ケーブルを推奨。NPLC＜1での電流測定は、まわりのノイズ環境を受ける。その為、測定確度を維持するために、適切なシールディングやガーディングを施す必要あり。

読み取り速度[7]

NPLC	アパーチャ	桁数	回／秒
0.0001	1.4µs	4.5	2,300
0.0006	10µs	5.5	1,350
0.01	167µs[8]	6.5	157
0.1	1.67ms[8]	6.5	108
1	16.6ms[8]	7.5	26
10	0.166s[8]	7.5	3
100		7.5	18/min

7. PRESET, DELAY 0, DISP OFF, OFORMAT DINTの場合。
8. アパーチャ時間は LFREQ に独立して設定可能。これらのアパーチャ時間は60Hzに対する時間。NPLCの値は1NPLC＝1/LFREQ。50Hzでは、アパーチャ時間を1.2倍、読み取り速度を0.833倍する。

最大入力

	定格入力	非破壊入力
I-LO間	±1.5Apk	<1.25Arms
LO-Guard間	±200Vpk	±350Vpk
Guard-Earth間	±500Vpk	±1000Vpk

4 / 交流電圧測定

一般事項

3458Aの真の実効値交流電圧測定には3つの方法があり、SETACVコマンドを使用して測定法を選択できます。以下に3種類の動作モードについての簡単な説明が記述されています。お客様の要求に合う測定法をお選びください。

次のセクションでは、これらの3つの動作モードを簡単に説明します。また、特定の測定ニーズに最適な方法を選択するために役立つ要約表も示します。

SETACV SYNC　Synchronously Sub-sampled Computed true rms technique.

このモードは、くり返し信号をサンプリングした後、計算により真の実効値を求めており、優れた直線性と**最も高確度な測定結果**を得ることができます。入力信号として繰り返し信号が必要です。このモードの周波数帯域は1Hzから10MHzです。

SETACV ANA　Analog Computing true rms conversion technique.

このモードは、電源投入時またはRESET時に選択されるデフォルト・モードで、10Hzから2MHz信号の真のRMS値を測定します。3つのモードのうち、**最高速の測定**を可能にします。

SETACV RNDM　Random Sampled Computed true rms technique.

このモードは、入力信号をサンプリングした後、計算により真の実効値を求めており、優れた直線性を得ることができます。入力信号は繰り返しの必要がなく、**広帯域のノイズ測定**に適しています。帯域は20Hz から10MHz です。

選択表

方法	周波数範囲	最高確度	繰り返し信号の必要性	回／秒 最小	回／秒 最大
周期サンプリング	1Hz～10MHz	0.010%	○	0.025	10
アナログ	10Hz ～2MHz	0.03%	×	0.8	50
ランダム・サンプリング	20Hz～10MHz	0.1%	×	0.025	45

周期サンプリング・モード (ACVファンクション、SETACV SYNC)

レンジ	フル・スケール	分解能	入力抵抗	温度係数[1] (読み取り値の%＋レンジの%)/ °C
10mV	12.00000	10nV	1MΩ±15% (＜140pF)	0.003 + 0.02
100mV	120.00000	10nV	1MΩ±15% (＜140pF)	0.0025 + 0.0001 [2]
1V	1.2000000	100nV	1MΩ±15% (＜140pF)	0.0025 + 0.0001
10V	12.000000	1μV	1MΩ±2% (＜140pF)	0.0025 + 0.0001
100V	120.00000	10μV	1MΩ±2% (＜140pF)	0.0025 + 0.0001
1000V	700.0000	100μV	1MΩ±2% (＜140pF)	0.0025 + 0.0001

AC確度 [2]

24時間～2年間(読み取り値の%＋レンジの%)

レンジ	ACBAND≦2MHz 1Hz～[3] 40Hz	40Hz～[3] 1kHz	1kHz～[3] 20kHz	20kHz～[3] 50kHz	50kHz～ 100kHz	100kHz ～ 300kHz	300kHz～ 1MHz	1MHz～ 2MHz
10mV	0.03 + 0.03	0.02 + 0.011	0.03 + 0.011	0.1 + 0.011	0.5 + 0.011	4.0 + 0.02		
100mV～10V	0.007 + 0.004	0.007 + 0.002	0.014 + 0.002	0.03 + 0.002	0.08 + 0.002	0.3 + 0.01	1 + 0.01	1.5 + 0.01
100V	0.02 + 0.004	0.02 + 0.002	0.02 + 0.002	0.035 + 0.002	0.12 + 0.002	0.4 + 0.01	1.5 + 0.01	
1000V	0.04 + 0.004	0.04 + 0.002	0.06 + 0.002	0.12 + 0.002	0.3 + 0.002			

1. Tcal ± 1ºC からの追加誤差、ただしACAL後±5ºC以内。

   ACBAND＞2MHzでは、すべてのレンジで10mV レンジの温度係数を使用。
2. 確度はフル・スケールからフル・スケールの10%、DC成分＜AC成分の10%、正弦波入力、クレスト・ファクタ＝1.4、PRESETとして仕様化されている。ACAL後24時間以内かつ±1ºC以内。

   (AC ＋ DC) 入力ピークは1000V レンジを除くすべてのレンジにおいて、フル・スケール×5 以下。

   NBS(米国標準局)に対するAgilent工場のトレーサビリティは、読み取り値の2ppmの追加誤差を加算。
3. LFILTER ONを推奨

AC確度(続き): 24時間〜2年間(読み取り値の%＋レンジの%)

	ACBAND ＞ 2MHz				
レンジ	45Hz〜100kHz	100kHz〜1MHz	1MHz〜4MHz	4MHz〜8MHz	8MHz〜10MHz
10mV	0.09 + 0.06	1.2 + 0.05	7 + 0.07	20 + 0.08	
100mV〜10V	0.09 + 0.06	2.0 + 0.05	4 + 0.07	4 + 0.08	15 + 0.1
100V	0.12 + 0.002				
1000V	0.3 + 0.01				

トランスファ確度

レンジ	読み取り値の%
100mV〜100V	(0.002 + %分解能) [1]

状態

- 4時間のフォーム・アップ後
- 基準測定から10分間、±0.5ºC
- 45Hzから20kHz、正弦波入力
- 基準電圧、周波数の±10%以内

1. % 単位の分解能は、RES コマンドまたはパラメータの値です(読み取り分解能は測定レンジのパーセント)。
2. Tcal ± 1ºC からの追加誤差、ただしACAL後±5ºC以内、(レンジの%)/ ºC。ACBAND ＞2MHz では、10mV レンジの温度係数を使用。

AC + DC確度(ACDCVファンクション)

AC＋DC確度はAC確度に以下の追加誤差(レンジの%)を加算。

DC成分＞AC成分の10%

レンジ	ACBAND≦ 2MHz	ACBAND＞2MHz	温度係数 [2]
10mV	0.09	0.09	0.03
100mV〜1000V	0.008	0.09	0.0025

DC成分＜AC成分の10%

レンジ	ACBAND≦ 2MHz	ACBAND＞2MHz	温度係数 [2]
10mV	0.7	0.7	0.18
100mV〜1000V	0.07	0.7	0.025

追加誤差

特定の測定セットアップに適合させるため、以下の追加誤差(レンジの%)を加算。

信号源	入力周波数 [3]			
インピーダンス	0〜1MHz	1〜4MHz	4〜8MHz	8〜10MHz
0Ω	0	2	5	5
50Ωターミネーション	0.003	0	0	0
75Ωターミネーション	0.004	2	5	5
50Ω	0.005	3	7	10

クレスト・ファクタ	乗数 [1]
1〜2	(%分解能)×1
2〜3	(%分解能)×2
3〜4	(%分解能)×3
4〜5	(%分解能)×5

3. フラットネス誤差には測定負荷が含まれる。

読み取り速度 [4]

ACBAND Low	最大(秒／回)
1〜5Hz	6.5
5〜20Hz	2.0
20〜100Hz	1.2
100〜500Hz	0.32
＞500Hz	0.02

%分解能最大	最大(秒／回)
0.001〜0.005	32
0.005 〜0.01	6.5
0.01〜0.05	3.2
0.05〜0.1	0.64
0.1〜1	0.32
＞1	0.1

4. この表は、最も遅い読み取り速度(秒／回)を示しており、実際の測定ではより高速に読み取れる場合がある。DELAY–1; ARANGE OFF実行時

セトリング特性

なし

コモン・モード除去比

(Lo端子に1kΩを接続した不平衡入力時) ＞90dB, DC〜60Hz

高周波温度係数

Tcal±5ºCからの追加誤差(レンジの%)/ ºC

	周波数	
レンジ	2～4MHz	4～10MHz
10mV～1V	0.02	0.08
10V～1000V	0.08	0.08

最大入力

	定格入力	非破壊入力
HI-LO間	±1000Vpk	±1200Vpk
LO-Guard間	±200Vpk	±350Vpk
Guard-Earth間	±500Vpk	±1000Vpk
VoltとHzの積	1×10^{8}	

アナログ・モード (ACVファンクション、SETACV ANA)

レンジ	フル・スケール	最大分解能	入力抵抗	温度係数 [1] (読み取り値の%＋レンジの%)/ °C
10mV	12.00000	10nV	1MΩ±15% (＜140pF)	0.003 + 0.006
100mV	120.0000	100nV	1MΩ±15% (＜140pF)	0.002 + 0
1V	1.200000	1µV	1MΩ±15% (＜140pF)	0.002 + 0
10V	12.00000	10µV	1MΩ±2% (＜140pF)	0.002 + 0
100V	120.0000	100µV	1MΩ±2% (＜140pF)	0.002 + 0
1000V	700.000	1mV	1MΩ±2% (＜140pF)	0.002 + 0

1. Tcal±1ºCからの追加誤差、ただしACAL後±5ºC以内。
2. 確度は、フル・スケールからフル・スケールの1／20、正弦波入力、クレスト・ファクタ＝1.4、PRESETとして仕様化されている。ACAL後、24時間以内かつ±1ºC以内。

   最大DCはACVファンクションでは400V以下。

   NBS(米国標準局)に対するAgilent工場のトレーサビリティは、読み取り値の2ppmの追加誤差を加算。

AC確度 [2]

24時間～2年間(読み取り値の%＋レンジの%)

レンジ	10Hz～20Hz	20Hz～40Hz	40 Hz～100Hz	100 Hz～20kHz	20kHz～50kHz	50kHz～100kHz	100kHz～250kHz	250kHz～500kHz	500 kHz～1 MHz	1 MHz～2 MHz
10mV	0.4 + 0.32	0.15 +0.25	0.06 +0.25	0.02 + 0.25	0.15 + 0.25	0.7 + 0.35	4 + 0.7			
100mV～10V	0.4 + 0.02	0.15 +0.02	0.06 + 0.01	0.02 + 0.01	0.15 + 0.04	0.6 + 0.08	2 + 0.5	3 + 0.6	5+2	10+5
100V	0.4 + 0.02	0.15 +0.02	0.06 + 0.01	0.03 + 0.01	0.15 + 0.04	0.6 + 0.08	2 + 0.5	3 + 0.6	5+2	
1000V	0.42+0.03	0.17 +0.03	0.08 + 0.02	0.06 + 0.02	0.15 + 0.04	0.6 + 0.2				

AC＋DC確度(ACDCVファクション)

AC＋DC確度はAC確度に以下の追加誤差(読み取り値の%＋レンジの%)を加算。

	DC成分≦AC成分の10%		DC成分＞AC成分の10%	
レンジ	確度	温度係数 [3]	確度	温度係数 [3]
10mV	0.0 + 0.2	0 + 0.015	0.15 + 3	0 + 0.06
100mV～1000V	0.0 + 0.02	0 + 0.001	0.15 + 0.25	0 + 0.007

3. Tcal±1ºCからの追加誤差、ただしACAL後±5ºC以内(読み取り値の%＋レンジの%)/ ºC。

追加誤差

特定の測定セットアップに適合させるため、以下の追加誤差を加算。

低周波誤差(読み取り値の%)

	ACBAND Low		
周波数	10Hz～1kHz NPLC＞10	1～10kHz NPLC＞1	＞10kHz NPLC＞0.1
10～200Hz	0		
200～500Hz	0	0.15	
500～1kHz	0	0.015	0.9
1～2kHz	0	0	0.2
2～5kHz	0	0	0.05
5～10kHz	0	0	0.01

クレスト・ファクタ 誤差 (読み取り値の%)

クレスト・ファクタ	追加誤差
1～2	0
2～3	0.15
3～4	0.25
4～5	0.40

読み取り速度[1]

		最大(秒／回)	
ACBAND Low	NPLC	ACV	ACDCV
≧10Hz	10	1.2	1
≧1kHz	1	1	0.1
≧10kHz	0.1	1	0.02

1. DELAY–1、ARANGE OFF実行時。

   DELAY 0、NPLC 0.1では、500回／秒以上(未仕様)で読み取れる場合がある。

セトリング特性

デフォルトのDELAYを使用した場合、最初の測定値あるいはレンジ変更の誤差は、入力ステップの0.01%の追加誤差を加算。以下のデータはDELAY 0に適用。

ファンクション	ACBAND Low	DC成分	セトリング時間
ACV	≧10Hz	DC＜10%のAC	0.5秒(0.01%まで)
		DC＞10%のAC	0.9秒(0.01%まで)
ACDCV	10Hz～1kHz		0.5秒(0.01%まで)
	1kHz～10kHz		0.08秒(0.01%まで)
	≧10kHz		0.015秒(0.01%まで)

最大入力

	定格入力	非破壊入力
HI-LO間	±1000Vpk	±1200Vpk
LO-Guard間	±200Vpk	± 350Vpk
Guard-Earth間	±500Vpk	±1000Vpk
VoltとHzの積	1×10^{8}	

コモン・モード除去比

(Lo端子に1kΩを接続した不平衡入力時) ＞90dB、DC～60Hz

ランダム・サンプリング・モード(ACVファンクション、SETACV RNDM)

レンジ	フル・スケール	最大分解能	入力抵抗	(温度係数[2] 読み取り値の%＋レンジの%)/ ºC
10mV	12.000	1µV	1MΩ±15% (＜140pF)	0.002 + 0.02
100mV	120.00	10µV	1MΩ±15% (＜140pF)	0.001 + 0.0001
1V	1.2000	100µV	1MΩ±15% (＜140pF)	0.001 + 0.0001
10V	12.000	1mV	1MΩ±2% (＜140pF)	0.001 + 0.0001
100V	120.00	10mV	1MΩ±2% (＜140pF)	0.0015 + 0.0001
1000V	700.0	100mV	1MΩ±2% (＜140pF)	0.001 + 0.0001

2. Tcal±1ºCからの追加誤差、ただしACAL後±5ºC以内。

   ACBAND＞2MHzの場合、すべてのレンジで10mVレンジの温度係数を使用。

AC確度[3]

24時間～2年間(読み取り値の%＋レンジの%)

	ACBAND≦2MHz				ACBAND＞2MHz				
レンジ	20Hz ～ 100kHz	100kHz ～ 300kHz	300kHz ～ 1MHz	1MHz ～ 2MHz	20Hz ～ 100kHz	100kHz ～ 1MHz	1MHz ～ 4MHz	4MHz ～ 8MHz	8MHz ～ 10MHz
10mV	0.5+0.02	4+0.02			0.1+0.05	1.2+0.05	7 + 0.07	20 + 0.08	
100mV～10V	0.08+0.002	0.3+0.01	1+0.01	1.5+0.01	0.1 +0.05	2+0.05	4 + 0.07	4 + 0.08	15 + 0.1
100V	0.12+0.002	0.4+0.01	1.5+0.01		0.12+0.002				
1000V	0.3+0.01				0.3+0.01				

3. 確度は、フル・スケールからフル・スケールの5%、DC成分＜AC成分の10%、正弦波入力、クレスト・ファクタ＝1.4、PRESETとして仕様化。ACAL後24時間以内かつ±1ºC以内。

   NBS(米国標準局)に対する10VDCのAgilent工場のトレーサビリティは、読み取り値の2ppmの追加誤差を加算。

   最大DCはACVファンクションでは400V以下。

AC＋DC確度(ACDCVファンクション)

AC＋DC確度はAC確度に以下の追加誤差(レンジの%)を加算。

	DC成分≦AC成分の10%			DC成分＞AC成分の10%		
レンジ	ACBAND ≦2MHz	ACBAND ＞2MHz	温度係数 [1]	ACBAND ≦2MHz	ACBAND ＞2MHz	温度係数 [1]
10mV	0.09	0.09	0.03	0.7	0.7	0.18
100mV～1 kV	0.008	0.09	0.0025	0.07	0.7	0.025

追加誤差

特定の測定セットアップに適合させるため、以下の追加誤差(レンジの%)を加算。

	入力周波数 [2]			
信号源インピーダンス	0～1MHz	1～4MHz	4～8MHz	8～10MHz
0Ω	0	2	5	5
50Ωターミネーション	0.003	0	0	0
75Ωターミネーション	0.004	2	5	5
50Ω	0.005	3	7	10

クレスト・ファクタ	乗数
1～2	(%分解能)×1
2～3	(%分解能)×3
3～4	(%分解能)×5
4～5	(%分解能)×8

読み取り速度 [3]

	最大(秒／回)	
%分解能	ACV	ACDCV
0.1～ 0.2	40	39
0.2～0.4	11	9.6
0.4～0.6	2.7	2.4
0.6～1	1.4	1.1
1～2	0.8	0.5
2～5	0.4	0.1
＞5	0.32	0.022

高周波温度係数

Tcal±5ºCからの追加誤差(レンジの%)/ ºC

レンジ	2～4MHz	4～10MHz
10mV～1V	0.02	0.08
10V～1000V	0.08	0.08

セトリング特性

デフォルトのDELAYを使用した場合、最初の測定値あるいはレンジ変更の誤差は、入力ステップの0.01%の追加誤差を加算。以下のデータはDELAY 0に適用。

ファンクション	DC成分	セトリング時間
ACV	DC＜ACの10%	0.5秒(0.01%まで)
	DC＞ACの10%	0.9秒(0.01%まで)
ACDCV	なし	

コモン・モード除比

(Lo端子に1kΩを接続した不平衡入力時)＞90dB, DC～60Hz

最大入力

	定格入力	非破壊入力
HI-LO間	±1000Vpk	±1200Vpk
LO-Guard間	±200Vpk	±350Vpk
Guard-Earth間	±500Vpk	±1000Vpk
VoltとHzの積	1×10^8	

1. Tcal ± 1ºC からの追加誤差、ACAL後±5ºC以内。(読み取り値の%)/ ºC。
   ACBAND＞2MHzの場合、すべてのレンジで10mV レンジの温度係数を使用。
2. フラットネス誤差には測定負荷が含まれる。
3. DELAY–1、ARANGE OFF実行時。ACVでDELAY 0の場合、読み取り速度はACDCVと同一。

5 / 交流電流測定

交流電流測定(ACIおよびACDCIファンクション)

レンジ	フル・スケール	最大 分解能	シャント抵抗	負荷電圧	温度係数[1] (読み取り値の%＋レンジの%)/ °C
100µA	120.0000	100 pA	730Ω	0.1V	0.002+0
1mA	1.200000	1nA	100Ω	0.1V	0.002+0
10mA	12.00000	10nA	10Ω	0.1V	0.002+0
100mA	120.0000	100nA	1Ω	0.25V	0.002+0
1A	1.050000	1µA	0.1Ω	＜1.5V	0.002+0

1. Tcal ± 1ºC からの追加誤差、ただしACAL後±5ºC以内。

AC確度[2]

24時間～2年間(読み取り値の%＋レンジの%)

レンジ	10Hz～20Hz	20Hz～45Hz	45Hz～100Hz	100Hz～5kHz	5kHz～20kHz[3]	20kHz～50kHz[3]	50kHz～100kHz[3]
100µA[4]	0.4+0.03	0.15+0.03	0.06+0.03	0.06+0.03			
1mA～100mA	0.4+0.02	0.15+0.02	0.06+0.02	0.03+0.02	0.06+0.02	0.4 +0.04	0.55+0.15
1A	0.4+0.02	0.16+0.02	0.08+0.02	0.1+0.02	0.3+0.02	1+0.04	

2. 確度は、フル・スケールからフル・スケールの1/20、正弦波入力、クレスト・ファクタ＝1.4、PRESETとして仕様化。ACAL後24時間以内かつ±1ºC以内。

   NBS(米国標準局)に対するAgilent工場のトレーサビリティは読み取り値の5ppmを加算。トレーサビリティは10V、10kΩのトレーサビリティの合計。

3. 代表値。
4. 100µAレンジでは1kHz最大。

AC＋DC確度(ACDCI Function)

AC＋DC確度はAC確度に以下の追加誤差(読み取り値の%＋レンジの%)を加算。

DC≦ACの10% 確度	温度係数[5]	DC＞ACの10% 確度	温度係数[5]
0.005+0.02	0.0+.001	0.15+0.25	0.0+0.007

5. Tcal ± 1ºC からの追加誤差、ただしACAL後±5ºC以内。(読み取り値の%＋レンジの%)/ °C

追加誤差

特定の測定セットアップに適合させるため、以下の追加誤差(レンジの%)を加算。

低周波誤差(読み取り値の%)

周波数	ACBAND Low 10Hz～1kHz NPLC ＞10	1～10kHz NPLC ＞1	＞10kHz NPLC ＞0.1
10～200Hz	0		
200～500Hz	0	0.15	
500～1kHz	0	0.015	0.9
1～2kHz	0	0	0.2
2～5kHz	0	0	0.05
5～10kHz	0	0	0.01

クレスト・ファクタ誤差(読み取り値の%)

クレスト・ファクタ	追加誤差
1～2	0
2～3	0.15
3～4	0.25
4～5	0.40

読み取り速度[6]

ACBAND Low	NPLC	最大(秒／回) ACI	ACDCI
≧10Hz	10	1.2	1
≧1kHz	1	1	0.1
≧10kHz	0.1	1	0.02

6. DELAY–1、ARANGE OFFの場合。DELAY 0、NPLC 0.1 では、500回／秒以上(未仕様)で読み取れる場合がある。

セトリング特性

デフォルトのDELAYを使用した場合、最初の測定値あるいはレンジの変更の誤差は、100µAから100mAレンジでは入力ステップの0.01%の追加誤差を加える。1Aレンジでは0.05%を加算。
以下のデータはDELAY 0に適用。

ファンクション	ACBAND Low	DC成分	セトリング時間
ACI	≧10Hz	DC＜10% AC	0.5秒(0.01%まで)
		DC＞10% AC	0.9秒(0.01%まで)
ACDCI	10Hz～1kHz		0.5秒(0.01%まで)
	1kHz～10kHz		0.08秒(0.01%まで)
	≧10kHz		0.015秒(0.01%まで)

最大入力

	定格入力	非破壊入力
I-LO間	±1.5Apk	＜1.25Arms
LO-Guard間	±200Vpk	±350Vpk
Guard-Earth間	±500Vpk	±1000Vpk

6 / 周波数／周期測定

周波数／周期測定

	電圧(ACまたはDC結合) (ACVまたはACDCV) [1]	電流(ACまたはDC結合) (ACIまたはACDCI) [1]
周波数レンジ	1Hz～10MHz	1Hz～100kHz
周期レンジ	1sec～100ns	1sec～10µs
入力信号レンジ	700Vrms～1mVrms	1A rms～10µArms
入力抵抗	1MΩ±15% (＜140pF)	0.1～730Ω [2]

1. 周波数／周期測定の信号の種類および結合方法は、FSOURCEコマンドにより設定。
2. レンジに依存。ACIのレンジ・インピーダンスを参照。
3. ゲート時間は測定分解能に依存。
4. 固定レンジで仕様化されている最大入力の場合。オート・レンジの場合、ACBAND≧1kHzでは最大速度は30回／秒。

   実際の読み取り速度は、入力の1周期、選択したゲート時間、またはデフォルト読み取りタイムアウト1.2sのうちいずれか長いものです。

確度

レンジ	24時間～2年間 0°C～55°C
1Hz～40Hz 1s～25ms	読み取り値の0.05%
40Hz～10MHz 25ms～100ns	読み取り値の0.01%

読み取り速度

分解能	ゲート時間 [3]	回／秒 [4]
0.00001%	1s	0.95
＞0.0001%	100ms	9.6
＞0.001%	10ms	73
＞0.01%	1ms	215
＞0.1%	100µs	270

測定方法:

レシプロカル方式

タイム・ベース:

10MHz±0.01%(0ºC～55ºC)

レベル・トリガ:

レンジの±500%(5%ステップ)

トリガ・フィルタ:

選択可能75kHzロー・パス・トリガ・フィルタ

スロープ・トリガ:

立ち上がり、立ち下がり

7 / ディジタイズ

一般事項

信号をディジタイズする場合、3458Aには3種類の方法があります。以下には、各モードについての簡単な説明が記述されています。お客様のアプリケーションに最も適した方法をお選びください。

DCV

標準DCVモード

標準DCVモードでは、28ビット分解能、0.2回／秒から、16ビット分解能、100k回／秒までの速度で、信号をディジタイズします。サンプル・アパーチャ時間は、500n秒から1秒(分解能100n秒)で可変です。入力電圧レンジは100mVから1000Vフル・スケール、帯域幅は30kHzから150kHzで測定レンジに依存します。

DSDC

直接サンプリング・モード(DC結合)

DSAC

直接サンプリング・モード(AC結合)

直接サンプリング・モードでは、入力信号は2nsの固定アパーチャ時間を持つトラック／ホールド回路を通り、16ビット分解能でディジタイズされます。サンプリング・レートは、6000s／サンプルから20µs／サンプルで可変で、100ns分解能で設定できます。また、入力電圧レンジは10mV(ピーク)から1000V(ピーク)フル・スケールです。周波数帯域は12MHz 。

SSDC

サブ・サンプリング・モード(DC結合)

SSAC

サブ・サンプリング・モード(AC結合)

サブ・サンプリング・モードでは、繰り返し信号を2nsの固定アパーチャ時間を持つトラック／ホールド回路に通し、16ビット分解能で等価サンプリングを行ないます。等価サンプリング・レートは、6000s／サンプルから10ns／サンプルで可変で、10ns分解能を持ちます。サンプルされたデータは3458A内部で時間順に並び変えられ、GPIBに出力できます。入力電圧レンジは、10mV(ピーク)から1000V(ピーク)フル・スケールです。また、周波数帯域は12MHz。

ディジタイズ・モード

モード	ファンクション	周波数レンジ	最高確度	サンプリング・レート
標準	DCV	DC～150kHz	0.00005～0.01%	100kサンプル/s
ダイレクト・サンプリング	DSDC / DSAC	DC～12MHz	0.02%	50kサンプル/s
サブ・サンプリング	SSDC / SSAC	DC～12MHz	0.02%	100Mサンプル/s(等価)

標準DCVモード(DCVファンクション)

レンジ	入力抵抗	オフセット電圧 [1]	帯域幅(代表値)	セトリング時間 (ステップの0.01%まで)
100mV	＞10^{10}Ω	＜5µV	80kHz	50µs
1V	＞10^{10}Ω	＜5µV	150kHz	20µs
10V	＞10^{10}Ω	＜5µV	150kHz	20µs
100V	10MΩ	＜500µV	30kHz	200µs
1000V	10MΩ	＜500µV	30kHz	200µs

1. AZEROの±1ºC､あるいはACAL後24時間以内かつ±1ºC以内。

DCパフォーマンス

読み取り値の0.005%＋オフセット [1]

最高サンプリング・レート(詳細は直流電圧測定を参照)

回／秒	分解能	アパーチャ時間
100k	15ビット	0.8µs
100k	16ビット	1.4µs
50k	18ビット	6.0µs

サンプル・タイムベース

確度: 0.01%

ジッタ:＜100ps rms

外部トリガ

レイテンシ:＜175ns [2]

ジッタ:＜50ns rms

レベル・トリガ

レイテンシ:＜700ns

ジッタ:＜50ns rms

2. 複数台の3458A間では＜125ns

ダイナミック・パフォーマンス

100mV、1V、10Vレンジ; アパーチャ時間＝6µs

テスト	入力(2×フルスケールpk-pk)	結果
DFT-高調波	1kHz	＜–96dB
DFT-スプリアス	1kHz	＜–100dB
微分非直線性	dc	＜レンジの0.003%
S/N比	1kHz	＞96dB

直接サンプリング・モード(DSDC、DSAC、SSDCおよびSSACファンクション)

レンジ[1]	入力抵抗	オフセット電圧[2]	帯域幅(代表値)
10mV	1MΩ (140pF)	＜50µV	2MHz
100mV	1MΩ (140pF)	＜90µV	12MHz
1V	1MΩ (140pF)	＜800µV	12MHz
10V	1MΩ (140pF)	＜8mV	12MHz
100V	1MΩ (140pF)	＜80mV	12MHz[3]
1000V	1MΩ (140pF)	＜800mV	2MHz[3]

1. DSACおよびSSACでは、最大DC電圧は400Vに制限される。
2. ACAL ACV後24時間以内かつ±1ºC以内。
3. 積は 1×10^8 V-Hzに制限。

DCから20kHzのパフォーマンス

読み取り値の0.02%＋オフセット[2]

最高サンプリング・レート

ファンクション	回／秒	分解能
SSDC、SSAC	100M (等価)[4]	16ビット
DSDC、DSAC	50k	16ビット

4. 等価サンプリング・レートは、繰り返し入力信号のサブ・サンプリングに使用された最小の時間インクリメント(10ns)により決定される。
5. 複数台の3458A間では＜25ns 。

ダイナミック・パフォーマンス

100mV、1V、10Vレンジ; 50,000サンプル／秒

テスト	入力(2×フル・スケールpk–pk)	結果
DFT-高調波	20kHz	＜–90dB
DFT-高調波	1.005MHz	＜–60dB
DFT-スプリアス	20kHz	＜–90dB
微分非直線性	20kHz	レンジの＜0.005%
S/N比	20kHz	＞66dB

サンプル・タイムベース

確度: 0.01 %

ジッタ:＜100ps rms

外部トリガ

レイテンシ:＜125ns[5]

ジッタ:＜2ns rms

レベル・トリガ

レイテンシ:＜700ns

ジッタ:＜100ps(1MHzフル・スケール入力)

8 / システム仕様

ファンション・レンジ測定

新しい測定構成をGPIBを通して送り、読み取りにトリガをかけ、コントローラにデータを戻すのに必要な時間。
(PRESET FAST, DELAY 0, AZERO ON, OFORMAT SINT, INBUF ON実行時)

測定構成の変更	GPIB速度 [1]	サブプログラム速度
DCV≦10VからDCV≦10V	180/s	340/s
任意のDCV/OHMSから任意のDCV/OHMS	85/s	110/s
任意のDCV/OHMSから任意のDCV/OHMS(DEFEATON)	150/s	270/s
任意のDCIから任意のDCI	70/s	90/s
任意のACVあるいはACIから、任意のACVあるいはACI	75/s	90/s

1. 9000シリーズ350使用。
2. SINTはAPER≦10.8μsecに対して有効。

動作速度 [2]

条件	レート
DCVオートレンジ・レート(100mVから10V)	110/s
コマンドの変更の実行(CALL、OCOMPなど)	330/s
GPIB へ読み取り値を送る、ASCII	630/s
GPIBへ読み取り値を送る、DREAL	1000/s
GPIBへ読み取り値を送る、DINT	50,000/s
内部メモリへ読み取り値を送る、DINT	50,000/s
内部メモリからGPIBへ読み取り値を送る、DINT	50,000/s
GPIBへ読み取り値を送る、SINT	100,000/s
内部メモリへ読み取り値を送る、SINT	100,000/s
内部メモリからGPIBへ読み取り値を送る、SINT	100,000/s
内部トリガの最高読み取り速度	100,000/s
外部トリガの最高読み取り速度	100,000/s

メモリ

	標準		オプション001	
	読み取り値	バイト	読み取り値	バイト
読み取り値の保存(16ビット)	10,240	20k	+65,536	+128k
サブプログラムやステート用の不揮発性メモリ		14k		

遅延時間

確度	±0.01%±5ns
最大	6000s
分解能	10ns
ジッタ	50ns pk-pk

タイマ

確度	±0.01%±5ns
最大	6000s
分解能	100ns
ジッタ	＜100ps rms

9 / 比測定

比測定の種類 [1]

DCV / DCV	レシオ＝(入力電圧)／(基準電圧)
ACV / DCV	基準電圧:(Hiセンス-Lo間)－(Loセンス-Lo間)
ACDCV / DCV	基準信号のレンジ:±12V DC (オートレンジのみ)

1. すべてのSETACVが設定可。Loセンス-Lo間は±0.25Vに制限。

確度

± (入力誤差＋基準誤差)
入力誤差 = 入力信号測定ファンクション(DCV、ACV、ACDCV)の全体の誤差×1
リファレンス誤差 = DC基準信号の全体のレンジ誤差×1.5

10 / 演算機能

一般演算関数の仕様

演算機能には、リアル・タイム処理と後処理の2通りの処理方法があります。

演算機能の仕様には、測定値の誤差あるいはユーザの入力値の誤差は含まれません。入力値や出力値の範囲は1.0×10^{-37}から 1.0×10^{37}です。レンジ外の値の場合、ディスプレイにはOVLDおよび、GPIB に1×10^{38}を送ります。最小実行時間は、それぞれの読み取りが終了した後、1 つの演算を実行するのに必要となる時間です。

NULL:
X–OFFSET
最小実行時間 = 180µs

PERC:
100×(X–PERC) / PERC
最小実行時間 = 600µs

dB:
20×Log (X/REF)
最小実行時間 = 3.9ms

RMS:
単極ディジタル・フィルタ
入力のRMSを計算
最小実行時間 = 2.7ms

STAT:
サンプルからMEAN、SDEVを計算
(個体数はN-1)　NSAMP、UPPER、LOWERも累積
最小実行時間 = 900µs

CTHRM2K (FTHRM2K):
2.2kΩサーミスタ (Agilent 40653A)の
°C (°F) への温度変換
最小実行時間 = 160µs

CRTD85 (FRTD85):
100Ω、α = 0.00385(40654Aあるいは40654B)の
°C (°F) への温度変換
最小実行時間 = 160µs

SCALE:
(X–OFFSET) / SCALE
最小実行時間 = 500µs

PFAIL:
MIN、MAXレジスタに基づく
最小実行時間 = 160µs

dBm:
10×Log [(X^2/RES) /1mW]
最小実行時間 = 3.9ms

FILTER:
単極ディジタル・フィルタ
入力の指数加重平均
最小実行時間= 750µs

CTHRM (FTHRM):
5kΩサーミスタ (Agilent 40653B)の
°C (°F)への温度変換

CTHRM10K (FTHRM10K):
10kΩサーミスタ (Agilent 40653C)の
°C (°F)への温度変換
最小実行時間 = 160µs

CRTD92 (FRTD92):
100Ω RTD、α = 0.003916の
°C (°F)への温度変換
最小実行時間 = 160µs

11 / 一般仕様

動作環境
動作温度: 0°C〜55°C
動作場所: 室内のみ
動作高度: 2,000mまで
ポリューション・レーティング: IEC 664 Degree 2

動作温度範囲
40°Cで95% RHまで

物理特性
88.9mm(高)×425.5mm(幅)×502.9mm(奥)
重量: 12kg
出荷重量 14.8kg

保管温度
–40°C〜+ 75°C

ウォーム・アップ時間
仕様を満たすには4時間必要。

電源
100/120V、220/240V±10%
48〜66Hz、360〜420Hz (自動センス)
＜30W、＜80VA (ピーク)
ヒューズ: 1.5 @ 115V, 0.5 A @230 V

クリーニングの手引き
マルチメータをクリーニングするには、
水で軽く湿したきれいな布を使用してください。

保証期間
1年

入力端子
金めっきテルルリウム銅

入力リミット
入力HI-LO間: 300VAC最大(CAT II)

IEEE-488インタフェース
以下の標準に準拠:
IEEE-488.1インタフェース標準
IEEE-728コード/フォーマット標準
CIIL(オプション700)

Agilent 3458Aの付属品:
テスト・リード・セット(Agilent 34118A)
電源コード
ユーザーズ・ガイド
校正マニュアル
アセンブリ・レベル修理マニュアル
クイック・リファレンス・ガイド

フィールド取り付けキット	Agilent部品番号
オプション001　拡張読み取り値メモリ	03458-87901
オプション002　高安定基準	03458-80002
追加キーボード・オーバレイ(5個1組)	03458-84303

その他のドキュメント	Agilent文章番号
Product Note 3458A-1: スループットと読み取りレートの最適化	5953-7058
Product Note 3458A-2: 3458Aを使用した高分解能デジタイズ	5953-7059
Product Note 3458A-3: 3458Aの電子校正	5953-7060
追加マニュアル・セット	03458-90000

付録B　GPIBコマンド

付録B GPIBコマンド

概要

本付録に記載したBASIC言語のGPIBコマンドは、HPシリーズ200/300コンピュータ専用です。任意のIEEE-488コントローラでこれらのメッセージを送信することができますが、ここで示すものとは構文が異なる場合があります。IEEE-488用語は、個々のコマンド・タイトルの後の括弧の中に記載しています。構文と例のすべてで、インタフェース・セレクト・コードが7で、デバイス・アドレスが22であることを想定しています。表29は、マルチメータのGPIB機能を示します。

表29. GPIB機能

IEEE 488.1機能	コード	説明
ソース・ハンドシェーク	SH1	マルチメータによる複数行メッセージの正しい転送を可能にします。
アクセプタ・ハンドシェーク	AH1	マルチメータによる複数行メッセージの正しい受信を可能にします。
トーカ	T5	マルチメータが"トーカ"になることを可能にします。これはGPIBを使用してデータを送信できることを意味します。またマルチメータによる、シリアル・ポールへの応答も可能にします。
リスナ	L4	マルチメータが"リスナ"になることを可能にし、これによりGIPBを使用して情報を受信できるようになります。
サービス要求	SR1	マルチメータによるサービス要求のサービス・コントローラへの非同期送信を可能にします。
リモート／ローカル	RL1	GPIBまたは前面パネルによりマルチメータをプログラムできるようにします。
パラレル・ポール	PPO	機能なし。
デバイス・クリア	DC1	コントローラから発行されたDevice Clearコマンドによりマルチメータをクリアされたステートへ初期化できるようにします。
デバイス・トリガ	DT1	マルチメータをGPIBを使用してトリガできるようにします。
コントロール・ファンクション	CO	機能なし。
デバイス・エレクトロニクス	E2	マルチメータが使用する電気的ドライバを記述します(E2 = 3ステート、1Mバイト/s最大)。

ABORT 7 (IFC)

マルチメータのインタフェース回路をクリアします。

構文 **ABORT 7**

例
```
ABORT 7          ! マルチメータのインタフェース回路をクリアします。
```

CLEAR (DCLまたはSDC)

マルチメータをクリアして、コマンドを受信できる状態にします。CLEARコマンドは、以下を行います。

- 出力バッファをクリアします。
- 入力バッファをクリアします。
- サブプログラムの実行を中止します。
- ステータス・レジスタをクリアします(ビット4、5および6は、これらのビットを設定する条件がまだ存在する場合は、クリアされません)
- ディスプレイをクリアします。
- トリガをオフにします(前のトリガ・モードは、マルチメータ・コマンドを送信することで再開できます)。

構文 **CLEAR 7**
CLEAR 722

例
```
LEAR 7        ! バス上のすべてのデバイス (DCL) をクリアします (SELECT CODE 7)。
CLEAR 722     ! アドレス 22 のデバイス (SDC) をクリアします (SELECT CODE 7)。
```

LOCAL (GTL)

マルチメータをリモート・ステートから解除し、そのキーボードをオンにします(マルチメータのLOCKコマンドによりキーボードがオフされていないことが条件です)。

構文 **LOCAL 7**
LOCAL 722

備考
- マルチメータのLOCALキーは、LOCAL LOCKOUTによりオフされます。LOCAL 722コマンドはキーボードをオンにしますが、その後のリモート・コマンドでキーボードはオフになります。ただし、LOCAL 7コマンドを送ると、その後のリモート・メッセージの後であっても前面パネル制御に戻ります。

例 `LOCAL 7` !GPIB REN ラインを FALSE に設定します (すべてのデバイスがローカルになります)
(ここで、リモート・モードに戻るには REMOTE 7 を実行する必要があります)。
`LOCAL 722` !アドレス 22 のデバイスに対して GPIB GTL を発行します (後で、任意のマルチメータ・コマンドまたは REMOTE 722 を実行すると、マルチメータはリモート・モードに戻ります)。

LOCAL LOCKOUT (LLO)

マルチメータのLOCALキーをオフにします。

構文 **LOCAL LOCKOUT 7**

備考
- LOCAL LOCKOUTを送ったときにマルチメータがローカル・ステートの場合は、ローカルのままになります。LOCAL LOCKOUTを送ったときにマルチメータがリモート・ステートの場合には、マルチメータのLOCALキーとキーボードが直ちにオフされます。
- LOCAL LOCKOUTを使用してLOCALキーをオフにすると、GPIB LOCAL 7コマンドを送るか、電源を入れ直さないとLOCALキーをオンにできません。マルチメータのLOCALキーがLOCAL LOCKOUTによりオフされている場合は、LOCAL 722コマンドによりキーボードはオンになりますが、その後のリモート・コマンドでキーボードはオフになります。ただし、LOCAL 7を送るとLOCALキーがオンになり、その後のリモート・メッセージの後でもオンのままになります。
- マルチメータのキーボードがLOCAL LOCKOUTコマンドとLOCKコマンドの両方でオフされている場合、キーボードの制御を戻すには、両方のコマンドをクリアする必要があります。LOCKは、LOCKをOFFに設定することでクリアできます。

例
```
10 REMOTE 722          !アドレス 22 のデバイスをリモート・ステートに設定します。
20 LOCAL LOCKOUT 7     !LOCAL LOCKOUT(LLO) をバス上の
30 END                 !すべてのデバイスに送ります。
```

REMOTE

GPIB RENラインを真(TRUE)に設定します。

構文 **REMOTE 7**
REMOTE 722

備考
- REMOTE 722コマンドにより、マルチメータがリモート・ステートになります。REMOTE 7コマンドそれ自体では、マルチメータはリモート・ステートにはなりません。REMOTE 7コマンドを送った後、マルチメータはリスン・アドレスを受信して初めてリモート・ステートになります。
- ほとんどの場合、LOCALコマンドを使用した後に必要なのはREMOTEコマンドだけです。REMOTEは、他のGPIBアクティビティに依存せず、RENと呼ばれるシングル・バス・ラインに送られます。ほとんどのコントローラが、電源を投入したとき、またはリセットされたときにRENラインを真(TRUE)に設定します。

例 `REMOTE 7` !GPIB REN ラインを TRUE に設定します。

上の行自体では、マルチメータはリモート・ステートになりません。マルチメータは、リスン・アドレス(例えば、OUTPUT 722;"BEEP"の送信)を受信して初めてリモート・ステートになります。

`REMOTE 722` !REN ラインを TRUE に設定して、デバイス 22 のアドレスを設定します。

上の行により、マルチメータがリモート・ステートになります。

SPOLL (シリアル・ポール)

SPOLLコマンドは、STB?コマンド(マルチメータ・コマンド・セット)と同様に、ステータス・ビット内の設定されたビットを表す数値(ステータス・バイト)を返します。返される数値は、設定されたすべてのビットの重み和です。

構文 **P=SPOLL (722)**

ステータス・レジスタ・ビット ビットと対応する重みは次のとおりです。

ビット番号	10進重み	説明
0	1	サブプログラム実行完了
1	2	上限または下限超過
2	4	SRQコマンド実行
3	8	電源投入時SRQ発生
4	16	命令準備完了
5	32	エラー (エラー・レジスタを確認)
6	64	サービス要求
7	128	データ使用可能

備考
- SPOLLを送ったときにSRQ行が真(TRUE)に設定されている場合は、ステータス・レジスタのすべてのビットがクリアされます(セットされた条件が存在する場合)。SPOLLを送ったときにSRQ行が偽(FALSE)の場合は、ステータス・レジスタの内容は変更されません。
- SPOLLコマンドとSTB?コマンドは、STB?がマルチメータのマイクロプロセッサを中断する点が異なります。したがって、STB?の場合、マルチメータが常に一見ビジーの状態になります(ビット4クリア)。SPOLLは、マイクロプロセッサを中断しないで単にステータス・バイトを抽出するだけです。したがって、SPOLLを使用すれば、マルチメータが別の命令を実行できる状態かどうかをモニタすることができます。
- SPOLL コマンドを送ったときにデータが出力バッファ内にある場合、データはそのままの状態に保たれます。ただし、STB?コマンドを送ったときにデータが出力バッファ内にある場合は、そのデータはステータス・データに置換されます。

例
```
10 P=SPOLL (722)          ! シリアル・ポールを送り、応答を P に入れます。
20 DISP P                 ! 応答を表示します。
30 END
```

TRIGGER (GET)

トリガがアーミングされると(TARMコマンドを参照)、TRIGGERコマンド(Group Execute Trigger)がマルチメータを1回トリガした後、トリガをホールドします。

構文 **TRIGGER 7**
TRIGGER 722

備考
- TRIGGERコマンドは、TRIG SGLコマンドが実行されたかのように、シングル・トリガを生成します。ただし、トリガがアーミング(TARMコマンド)されていなければマルチメータはトリガされません。
- PAUSEコマンド(マルチメータ・コマンド・セット)によってサブプログラム・メモリの実行が一時停止されている場合、TRIGGERコマンドがサブプログラムの実行を再開しますが、シングル・トリガは生成しません。

例
`TRIGGER 7` !GROUP EXECUTE TRIGGER(GET) を送ります。
`TRIGGER 722` !GROUP EXECUTE TRIGGER(GET) をアドレス 22 のデバイスに送ります。

付録C　前面/裏面端子スイッチと ガード端子スイッチのロック

付録C　前面/裏面端子スイッチと ガード端子スイッチのロック

はじめに

前面/裏面端子スイッチとガード端子スイッチは、設定の変更を防ぐために、それぞれ、または両方をロックすることができます。ロックするには、まず3458からすべてのカバーを取り外します。次に、前面/裏面端子スイッチ、ガード端子スイッチのプッシュロッドを取り外します。そして、プッシュロッドが出ていた穴にスイッチ・カバーを取り付けます。スイッチ・カバーは、前面/裏面端子およびガード端子スイッチ・ロックアウト・キットに入っています。最後に、カバーを取り付けます。

警告　**この手順は、修理資格のあるサービスマンだけが行なってください。怪我を避けるため、サービスマン以外は以下の作業を行わないでください。**

必要なツール

1. #1 Pozidrivドライバ
2. #TX 15 Torxドライバ
3. #TX10 Torxドライバ

手順

ロックアウト・キットのインストール手順は、次のステップから構成されます。

- カバーの取り外し
- ガード端子スイッチのプッシュロッドの取り外し
- 前面/裏面端子スイッチのプッシュロッドの取り外し
- スイッチ・キャップの取り付け
- カバーの取り付け

カバーの取り外し

以下を行います。

1. 3458からすべての接続を外します。
2. 3458のAC電源コードを外します。
3. 図35を参照してください。3458の(前面パネルから見て)右面を前にして置きます。

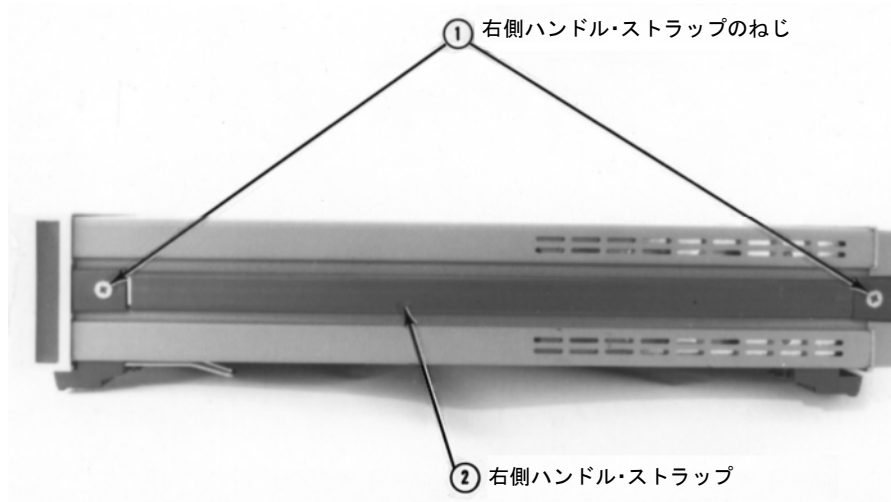



図35. 3458の右面図

4. #l Pozidrivを使用して、右側ハンドル・ストラップのねじを外します。右側ハンドル・ストラップを取り外します。
5. 図36を参照してください。3458の左面を前にして置きます。
6. #l Pozidrivを使用して、左側ハンドル・ストラップのねじを外します。左ハンドル・ストラップを取り外します。
7. #TX10 Torxドライバを使用して、上面および底面カバーのグランドねじを外します。図37を参照してください。
8. 図38を参照してください。3458の裏面を前にして置きます。
9. #TX15 Torxドライバを使用して、裏面ベゼルのねじを外します。裏面ベゼルを取り外します。
10. 上面カバーを取り外します。カバーを裏面方向に引いてから、取り外します。
11. 上面が下になるように、3458を天地逆にして置きます。底面カバーを取り外します。カバーを裏面方向に引いてから、取り外します。そのままの状態で、次のステップを行います。

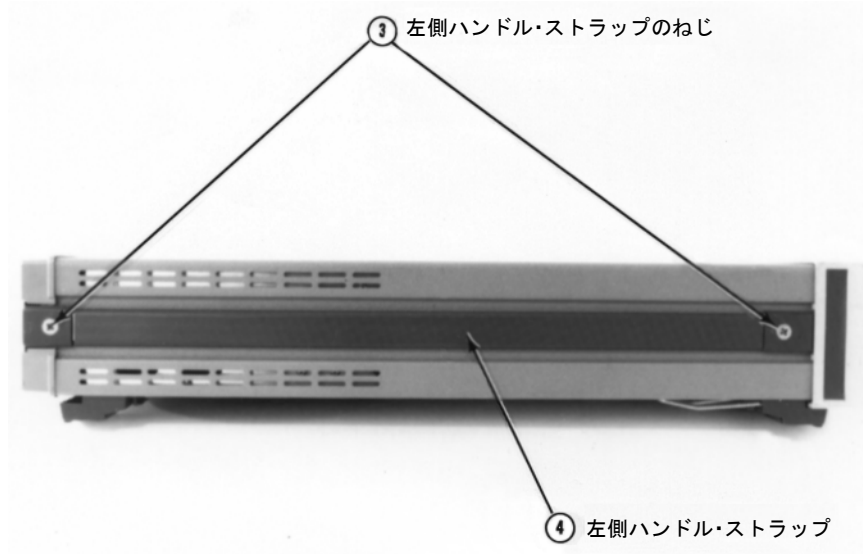



図36. 3458の左面図

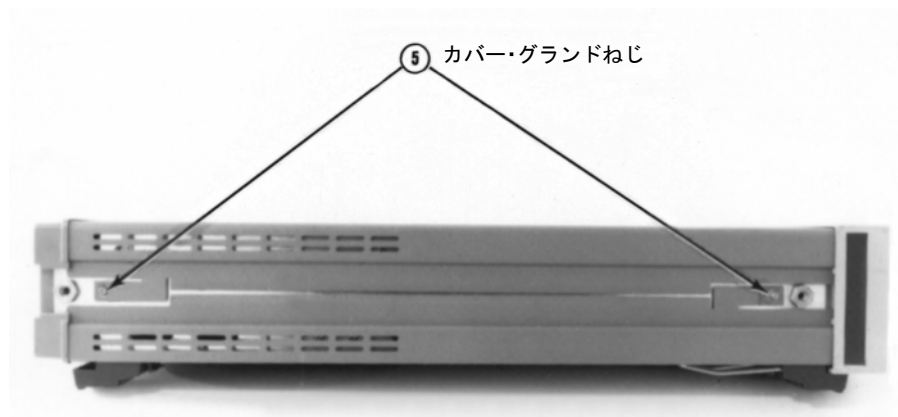



図37. カバー・グランドねじ

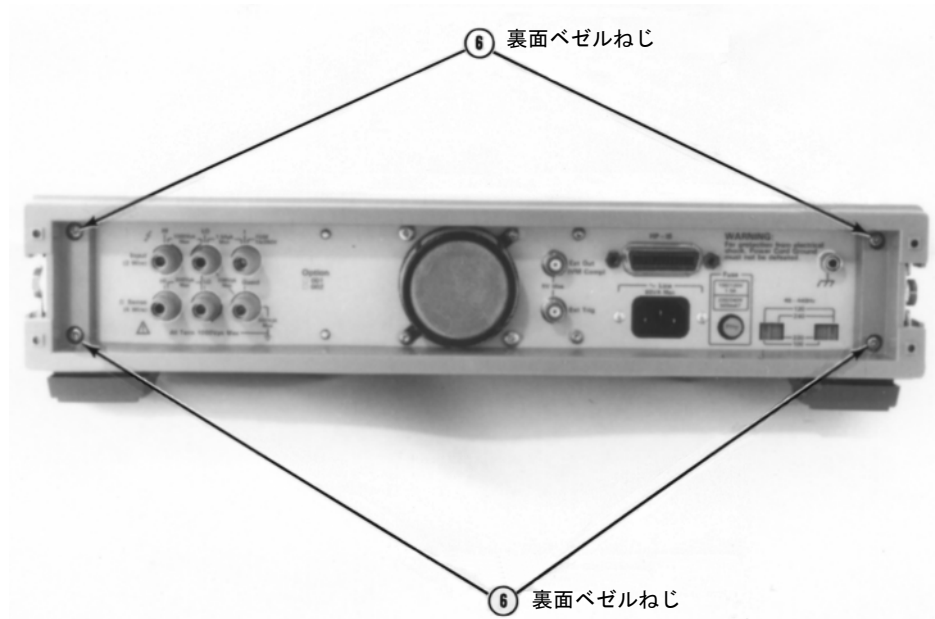



図38. 3458の裏面図

ガード端子スイッチのプッシュロッドの取り外し

ガード端子スイッチをロックしない場合は、このステップを飛ばしてください。

1. 図39を参照してください。#TX10 Torxドライバを使用して、底面シールドのねじを外します。次に、底面シールドを取り外します。シールドを裏面方向に引いて、リテーナがシールドのスロットと合うようにしてから、シールドを外します。

2. 図40を参照してください。ガード端子スイッチのプッシュロッドを取り外します。外しにくいときは、小型のマイナス・ドライバでこじ開けて取り外してください。スイッチを必要な位置に合わせます。

3. 図39を参照してください。底面シールドを取り付けます。シールドのスロットを、リテーナに合わせます。シールドを前面パネル方向に押して、シールドのねじ穴とシャーシのねじ穴を合わせます。#TX10 Torxドライバを使用して、シールドねじを元に戻します。

前面/裏面端子スイッチのプッシュロッドの取り外し

前面/裏面端子スイッチをロックしない場合は、このステップを飛ばしてください。

1. 図41を参照してください。底面が下になるようにして3458を置きます。

2. #TX10 Torxドライバを使用して、上面シールドのねじを外します。次に、上面シールドを取り外します。シールドを裏面方向に引いて、リテーナがシールドのスロットと合うようにしてから、シールドを外します。

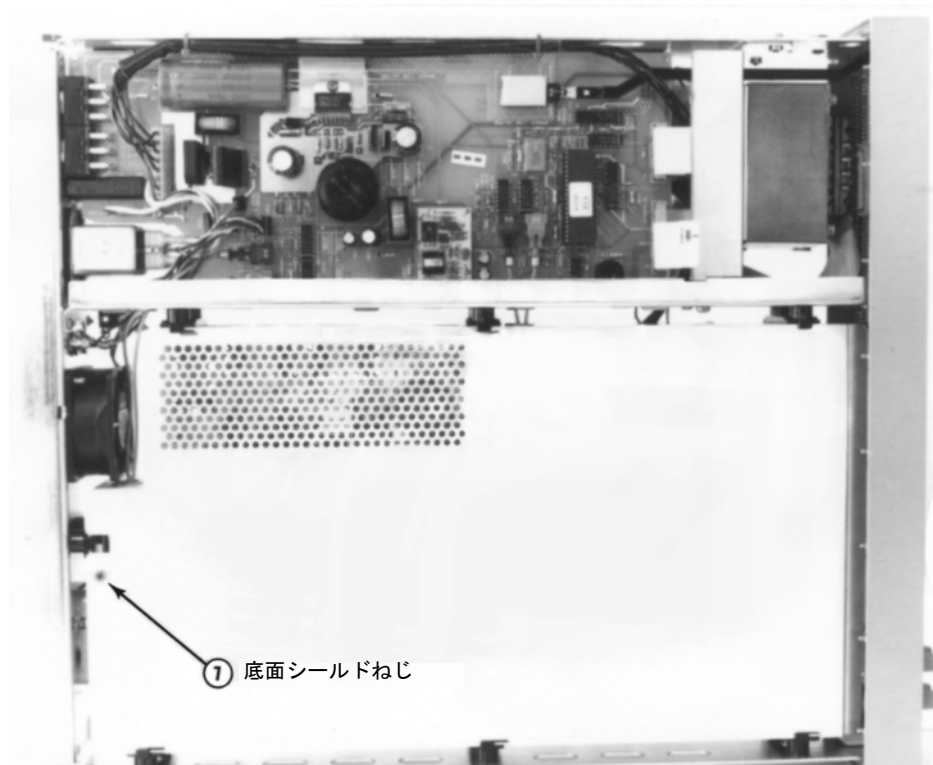



図39. 3458内部底面図

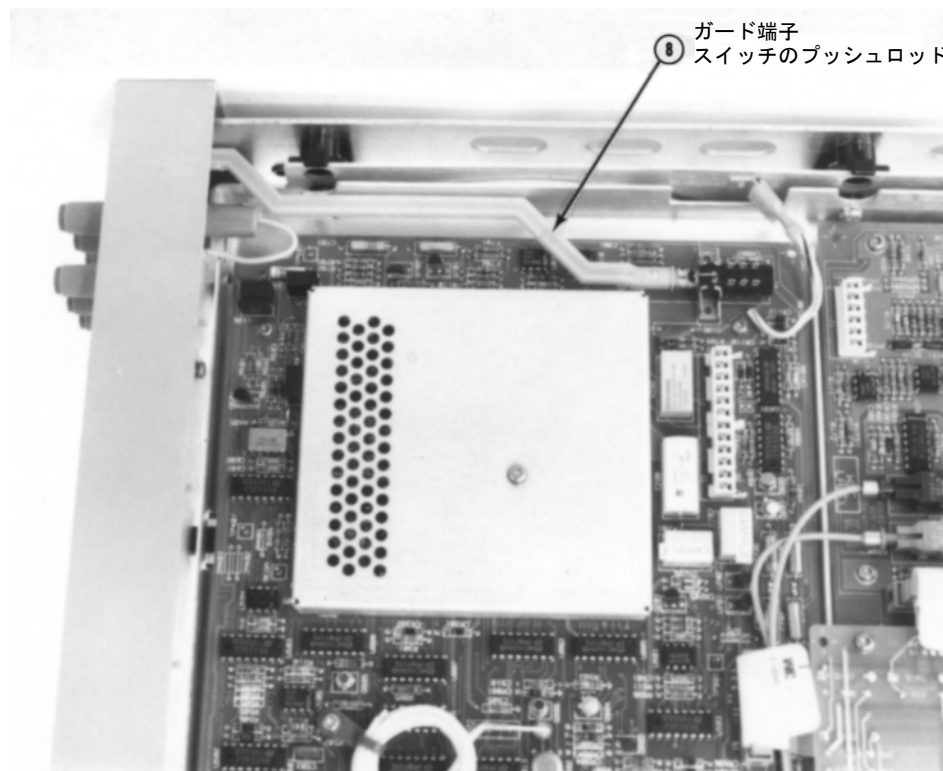



図40. ガード端子スイッチとプッシュロッド

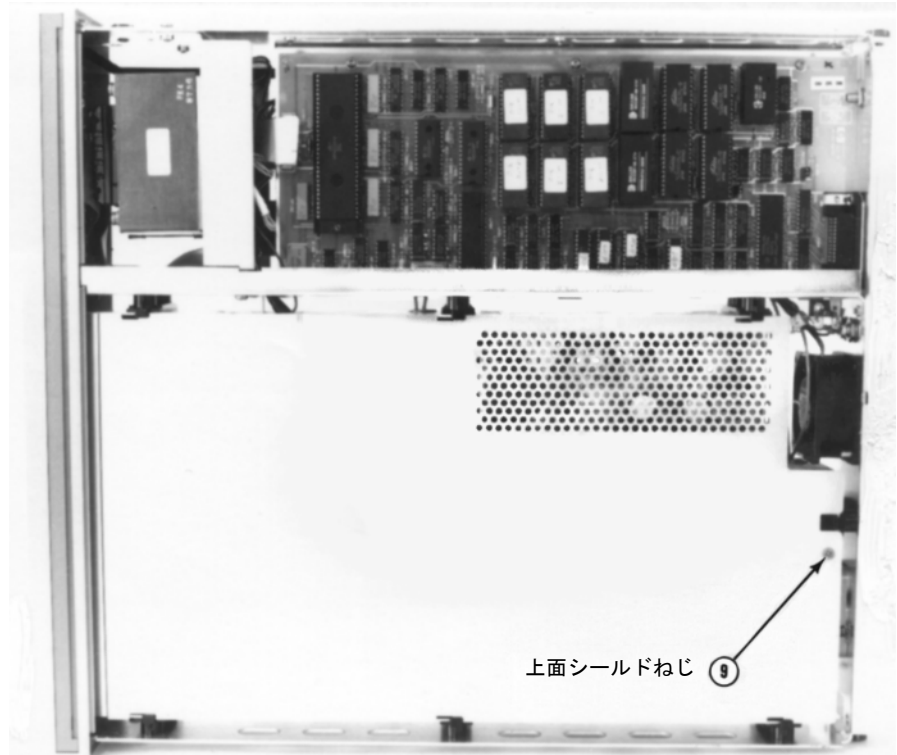



図41. 3458内部上面図

3. 図42を参照してください。前面/裏面端子スイッチのプッシュロッドを取り外します。外しにくいときは、小型のマイナス・ドライバでこじ開けて取り外してください。スイッチを必要な位置に合わせます。

4. 図42を参照してください。上面シールドを取り付けます。シールドのスロットを、リテーナに合わせます。シールドをフロントパネル方向に押して、シールドのねじ穴とシャーシのねじ穴を合わせます。#TX10 Torxドライバを使用して、シールドねじで取り付けます。

スイッチ・キャップの取り付け

以下を行います。

1. 図43を参照してください。前面パネルを前にして置きます。

2. 前面/裏面端子スイッチ、およびガード端子スイッチ用の穴を確認します。

3. スイッチ・ロックアウト・キットにある、図43で示すような小型の四角いカバーを確認します。

4. このカバーのタブを、前面/裏面端子スイッチ、またはガード端子スイッチ穴の上下に合わせます。

5. タブをつまんで、カバーを止まるまでスイッチ穴に押し込みます。固定させます。

6. 必要に応じて、もう一方のスイッチ穴に対してもステップ4と5を行います。

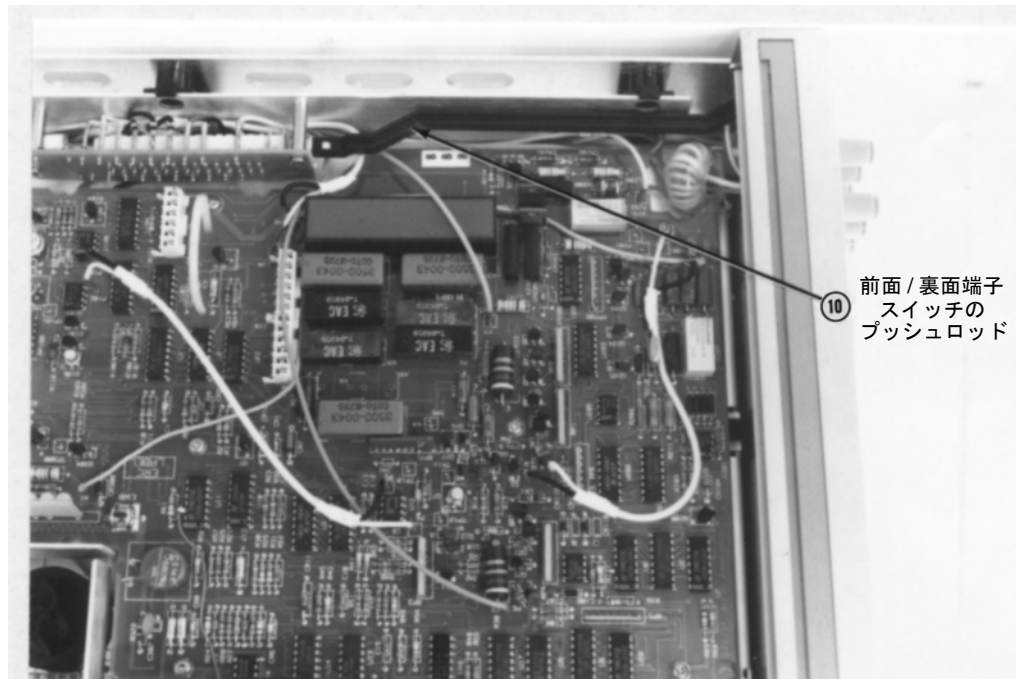



図42. 前面/裏面端子スイッチとプッシュロッド

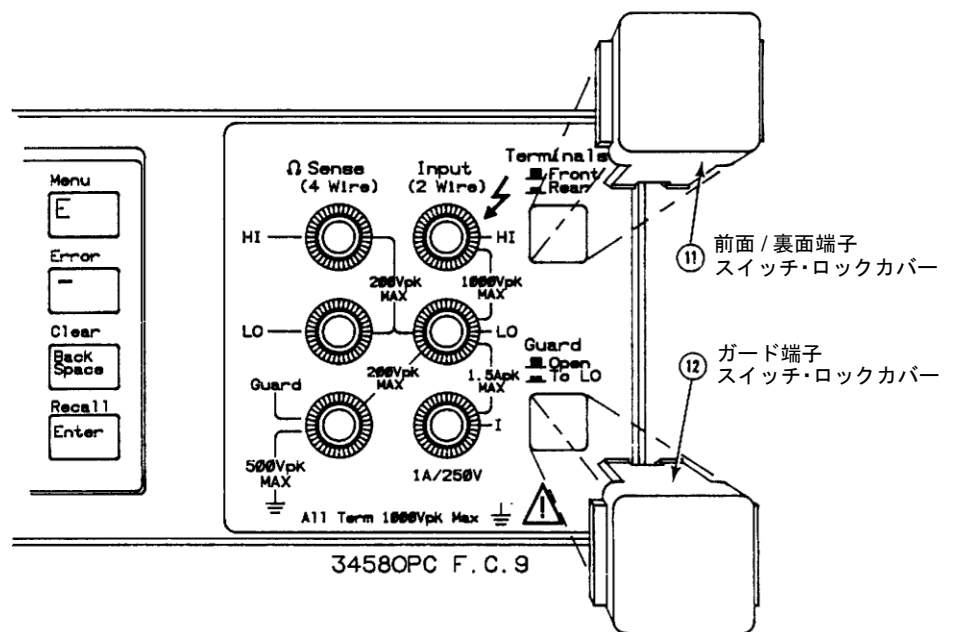



図43. スイッチ・カバーの取り付け

カバーの取り付け

次を行います。

1. 上面を下にして3458を置きます。
2. 底面カバーを、サイドのスロットにはめます。次に前面パネルの方向に押して、フロントパネル・ベゼルにはめます。
3. 底面を下にして3458を置きます。
4. 上面カバーを、サイドのスロットにはめます。次に前面パネルの方向に押して、フロントパネル・ベゼルにはめます。
5. 図38を参照してください。裏面を前にして3458を置きます。
6. 裏面ベゼルを元に戻します。#TX15 Torxドライバを使用して、裏面ベゼルねじを締めます。
7. 図37を参照してください。左面を前にして3458を置きます。#TX10 Torxドライバを使用して、上面および底面カバーのグランドねじを元に戻します。

警告　**安全と適正な動作のために、カバー・グランドねじは必ず取り付けてください。**

8. 図36を参照してください。左側ハンドル・ストラップを元に戻します。#1 Pozidrivドライバを使用して、ストラップねじを元に戻します。
9. 図35を参照してください。右面を前にして3458を置きます。
10. 右側ハンドル・ストラップを取り付けます。#1 Pozidrivドライバを使用して、ストラップねじを元に戻します。
11. これで3458を使用できます。電源を投入したら、自動校正の実行をお勧めします。自動校正の実行には、"ACAL ALL"コマンドを使用します。

付録D　スルー プットと測定速度の最適化

付録D　スループットと測定速度の最適化

(Product Note 3458A-1から)

過去15年間でマイクロコンピュータは、その内部速度、他の装置との通信速度ともに大きく向上しました。計測器で使用されるマイクロコンピュータの実際のクロック速度は1MHz以下から12MHz以上へと、またデータ・バスは8ビットから16ビットに進化しました。

また、同時期にシステム・マルチメータも、動作速度と測定速度の面でそれ以上といってよい発展を遂げました。1975年には、5 1/2桁の分解能で毎秒24回の測定は非常に高速とされていました。今日の3458Aマルチメータなら、5 1/2桁の分解能で毎秒50,000回の測定を行います。これは、実に2,000倍の速度の向上です。この驚くべき速度の向上は、単にマイクロコンピュータの高速化だけでなく、アナログ-ディジタル変換プロセスの進化、マイクロコンピュータの使用方法の改善、ユーザ・ニーズのより良き理解などに基づいています。

3458Aマルチメータについて

3458Aマルチメータの測定速度は、4 l/2桁DC電圧測定の毎秒100,000回から8 l/2桁DC電圧測定の毎秒6回までが可能で、高分解能ほど低速度というトレードオフによって、その間を選択することができます。またAC電圧測定のような従来は低速の測定も、3458Aは高速に行います。例えば10kHzを超える入力周波数に対し、最大毎秒50回の測定速度で高確度の真の実効値ACVを測定します。マルチメータの測定速度が増大すれば、テスト・スループットも向上することは明らかです。しかし、明らかではないが、スループットに大きく影響する要素として、ファンクション／レンジ／読み取り速度(積分時間)／インタフェース・モードを変更する際のマルチメータの動作速度があります。3458Aは、ファンクションおよびレンジの変更、測定の実行、結果の出力を、毎秒200回の速度で行います。

アプリケーション指向コマンド言語

ハードウェア依存度が少なくアプリケーション指向の測定を目指す試みは、マルチメータのコマンド言語を進化させ、簡単で使いやすいプログラミングを可能にしました。しかしこのような進化には、オーバヘッドの増加とコマンドへの応答の低速化が伴います。3458Aマルチメータは特にこの問題に対処するよう設計されていて、使いやすいアプリケーション指向コマンド言語を提供するとともに、コマンドへの高速応答を可能にしています。

本質的に遅い測定

ある種の測定は、本質的に高速な測定になじまないことが良く知られています。例えば高インピーダンスの測定、低周波イベントの周波数測定、実効値AC電圧/電流測定、またノイズ存在下での正確な測定などです。しかしその本質的な低速性にもかかわらず、それらの測定を必要とするテスト・システムで、スループットを実質的に増大させることも可能です。3458Aはテスト・スピードを増大させるさまざまな選択を可能にして、これを実現します。例えば多くのシステムで、確度を抑えて速度を増加させられます。また測定タイミングの柔軟性によって、確度を保ちながら実効値AC測定の速度を増すことも可能です。3458Aマルチメータで可能な、このようなトレードオフについては、このプロダクト・ノートで詳細に説明しています。

テスト速度の最大化

プログラム・メモリ

テスト・プロセスの速度は、3458Aとコンピュータ間の通信経路を調整することでも最大化できます。一般に、DMM(ディジタル・マルチメータ)はシステム内で最高速の装置です。したがって、一連の測定を行う場合に、コンピュータは他の装置との通信により多くの時間を割く必要があります。3458Aマルチメータは、測定タスクの割り当てをコンピュータとDMM間で最適に割り当てるための、いくつかの機能を持っています。3458Aのユニークな不揮発性プログラム・メモリにより、外部／補助／GPIB[1] トリガなどの外部イベントを使って、測定シーケンスの動的な進行を可能にします。また、プログラム・メモリを使用して測定シーケンスの完全なプログラム化が可能なので、コントローラなしのスタンドアローン動作として、前面パネルから実行することができます。

ステートの保存

ステートの保存を使うと、簡単なコマンドによって、静的な測定器ステートを完全に定義し、呼びだすことができます。また3458AはGPIB経由で、または自身の標準の10,000個(オプションで75,000個)の読み取り値メモリに対して、毎秒100,000回の測定速度で測定データを高速転送します。

読み取り値の解析

さらに3458Aは、内部でデータ解析を行うという機能も備えています。これにより、スループットを増大させるだけでなく、統計による品質管理やリミット・チェックに必要なデータも提供されます。プログラム・メモリは合否演算機能の実行が可能で、割り込みフラグをセットしてリミット外の測定をコンピュータに警告できます。または最大限の測定速度を保持しつつ、さまざまな演算機能を使用して、メモリ内のデータを後処理することも可能です。例えば統計機能(平均値、標準偏差、最大値、最小値、読み取り数)、サーミスタの線形化、RTDの線形化、スケーリング、フィルタ機能などが可能です。データ解析をコンピュータで行うか、DMMで実行するかは、それぞれのテスト・タスクや簡単なプログラミング・コマンドでこれらの解析機能を可能にする便利さに依存します。

タスクのグルーピングとシーケンシング

同様の測定同士をグループ化して測定器構成の数を少なくする、測定シーケンスの調整を行なっても、テスト・スループットを増大させることができます。自動プログラム・ジェネレータを使わずに、カスタムのプログラムを書けばこれが可能になります。プログラム・ジェネレータは一般にプログラミングのしやすさのために最適化されており、テスト・タスクのための最も簡単な方法を提供します。これにより、各テスト・グループのリミット選択は可能ですが、必ずしも最高速のスループットのためのグループ化はできません。FTM300などのファンクション・テスト管理ソフトウェアでは、テストをスループットのためにカスタマイズでき、その上で統計品質管理(SQC)や在庫管理などのオーバヘッド・プログラミングの70%を提供します。

1. GPIBはIEEE 488規格の実装で、ANSI MC1.1規格"Digital interface for programmable instrumentation"と同一のものです。

システムのアップタイム

システムのアップタイムが長いということも、高スループットを意味します。3458Aは高安定内部基準を使用して、ACを含む全ファンクションに対して完全な自己校正を行います。この自己(自動)校正により、時間ドリフトや、ラック内や卓上での温度変動による測定誤差を除去し、優れた確度が提供されます。外部基準を使用した定期校正が必要な場合も、精密10VDC信号源および精密10kΩ抵抗器を接続するだけで済みます。外部基準に対する精密な内分比転送測定によって、ACを含むすべてのファンクションとレンジが自動的に校正されます。(校正については、Product Note 3458A-3で詳述しています。)

システムのアップタイムは、そのコンポーネントの信頼性が向上することによっても増加します。3458Aの高い信頼性は、Agilentの欠陥低減"10X"運動の成果です。製品開発のデザイン段階での環境／誤使用／ストレス・テストによって、Agilentは欠陥と初期不良の数を、過去10年間にわたり10分の1に削減しました。

目的

このプロダクト・ノートの目的は、3458Aマルチメータのこれまでにない高速度と確度を使用して、ユーザのアプリケーションに対して最大限のテスト・スループットと測定速度を実現することです。これは、3458Aで可能なトレードオフを説明することにより、またHP 9000 シリーズ200/300コンピュータを併用することによって達成されます。

このプロダクト・ノートの内容

- DC測定(電圧、電流、抵抗) - 速度、分解能、確度の間で可能なトレードオフを、ユーザのテスト・システムの最適化にどう適用するか
- AC測定(アナログACV、同期ACV、ランダムACV、電流) - ユーザのアプリケーションに最適なモードと仕様の選択
- 周波数と周期 - 必要な速度、確度、分解能を得るためのゲート時間の選択
- タスク割り当てによるテスト・プロセスの最適化 - 内蔵の演算関数および後処理演算、読み取り値メモリ、ステート・メモリ、プログラム・メモリを使用した、DMMとコンピュータ間でのタスクの最適な割り当てと構成(サンプル・プログラムを使用)
- 最大スループットを得るための、合否リミット・チェックと統計機能を使用した、適切な構造化プログラムによるベンチマーキング

DC電圧、DC電流、抵抗

3458Aは、A/D変換器に直接向かう標準のDCV経路と、トラック・アンド・ホールド回路に向かう経路(トラック・アンド・ホールド経路)の、2つの測定経路を持っています。DCV経路の帯域幅は150kHzに制限されますが、トラック・アンド・ホールド経路は12MHzまでの信号を許容します。これに対し、繰り返し測定を実行しない限りトラック・アンド・ホールド経路の分解能は16ビットに限られますが、DCV経路は8 1/2桁(27ビット)の分解能を提供します。

DCV経路の最適化

3458Aで可能な従来のトレードオフは、測定速度と測定分解能のバランスです。デザインの初期段階で、入力経路の実数抵抗成分に関連した本質的なジョンソン・ノイズを低減しているため、積分測定の分解能は前世代のDMMに比較して3倍向上しています。例えば3457Aでは、積分周期が1電源ライン周波数(PLC)または17msで、6 1/2桁(3,000,000カウント)の測定が可能です。同じ積分周期で、3458Aなら7 l/2桁(12,000,000カウント)の測定を行います。同様に、優れた直線性を保証する特別な注意が払われており、直線性は3457Aに比較して10倍向上しています。これらの結果、これまでにない高速で正確な測定が実現されました。またこれは、確度と分解能の向上により、以前は10PLCかかった測定を3458Aでは1PLCで行えることを意味しています。

高速のみが重要な測定の場合、または電源ノイズが問題でない状態で分解能と確度をトレードオフする場合には、3458Aはアパーチャ 500nsにおける4 l/2桁から、アパーチャ 1秒での8 1/2桁までの性能を、100nsのステップで提供します。図44は、アパーチャと測定速度／ノイズ／分解能／確度との対応を示しています。

図44のグラフから、実際のアパーチャ (積分周期)の、読み取り速度に対する影響が分かります。ただしここでは、スループットに対するHPMLコマンドの影響、3458Aの基本的な動作原理のいくつかが隠されています。HPMLは、アプリケーション指向コマンド・セットです。このコマンド・セットの基本的な考え方は、どのような測定をするかさえ理解していれば、その測定のために3458Aが何をするかは知らなくてよいというものです。しかし、複雑なアプリケーションに対してスループットを向上させるには、単なる測定の実行だけでなく、3458Aの動作を理解する必要があります。ここで行うトレードオフは、確度や使い勝手の代りに速度を増加させることです。測定の観点から見て、スループット・スピードに最も影響するHPMLコマンドには次のものがあります。

```
FUNC<DCV, DCI, OHM, FOHM>,<range>,<resolution in %>
NPLC #
APER<integration period s>
RES <resolution in %>
AZERO,<on or off>
```

表30は、NPLCコマンドとAPERコマンドがいくらかの互換性を持つことを示しています。これらのコマンドの最大の違いとして、NPLCは選択した電源周波数の倍数／約数の積分周期を設定するのに、実際の電源周波数を使用します。これに対してAPERは、500nsから1sまで100nsのステップにより秒という基本単位で積分周期を設定します。例えば電源周波数60Hzで動作時に、NPLC 1はAPER 0.016666と同じです。

図44. アパーチャ(NPLC)に対する確度、測定速度、分解能、ノイズの依存度

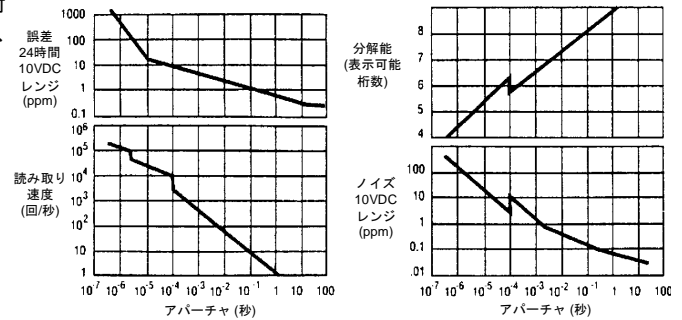



表30. 積分時間と問い合わせ応答

コマンド	積分時間(APER)		問い合わせ応答(NPLC?)	
	50 Hz	60 Hz	50 Hz	60 Hz
NPLC0	500 ns	500 ns	25 E-6	29.99994 E-6
NPLC.5	10ms	8.333 ms	500 E-3	499.99700 E-3
NPLC 1	20 ms	16.6667 ms	1	1
NPLC10	200 ms	166.667 ms	10	10
NPLC 11*	200 ms	166.667 ms	20	20
＊NPLC>10のとき、連続積分時間はNPLC 10の積分時間と同じですが、2回以上の読み取りが行われ、その平均の結果はディスプレイまたはGPIBに出力されます。				

NPLCが1から10の間のとき(1、10含む)、NPLCは次の整数に切り上げられます。NPLC>10のとき、NPLCの実際の値は次の10の倍数の整数に切り上げられます。NPLC<1のときは、電源ライン周波数で積分周期がスケーリングされることを除いて、選択した値がアパーチャとして使用されます。例えばNPLCを0.1PLCに設定すると、3458Aは実際には積分周期を0.1×(100nsステップに切り上げた電源ライン周期)、0.0016666s(60Hz動作)に設定します。問い合わせNPLC?は、99.9958E-3 PLCを返します。NPLCの値が2.5なら、3458Aは積分周期を3PLCに設定します。値が21なら、積分周期は30PLCとなります。NPLC 0のときは、常に最小の積分周期500nsまたは29.99994E-6 PLC(60Hz動作)となります。

積分周期に影響するコマンドとしては、他に分解能コマンドRESがあります。コマンドRESは、表示値の桁数を最大入力パラメータの%として設定します。測定の分解能は、ファンクション・コマンドの一部として、またはRESコマンドにより設定できます。これにより積分周期は、ADCが測定を要求された分解能に変換可能な値に設定されます。

例えば

```
DCV,20,.001  !(このコマンドの分解能パラメータを使用)
```

および

```
DCV,20;RES.001          !(DCVコマンドのパラメータを省略して、RESコマンドを使用)
```

両方とも3458AをDCVに、レンジ100V、積分周期8µs、分解能を20Vの0.00l%に設定します。また測定速度は、オートゼロ動作をオフにするだけで倍になります。オートゼロ・オン(AZERO,ON)は、3458Aのデフォルト設定です。この設定では、3458Aの入力端子の熱起電力(オフセット電圧)を除去するために、内部で入力をショートさせてオフセット電圧を測ります。そして測定したDCオフセットを実際の入力電圧から減算して、最終的な出力として示します。したがって通常は、実際には1つの測定に2回の測定サイクルが実行されています。このプロセスは3458Aの確度仕様を保証するものですが、単に入力電圧を測る場合の半分の速度しか実現しません。温度が安定した環境では、この機能をオフにしても、短い時間(10分くらい)なら非常に小さな確度の減少しか起りません。このため、積分時間の短縮の他に、AZERO OFFは測定速度の向上に使える最も重要なコマンドとなります。

DC電流

電流測定にも、DCV測定の場合と同様の説明を適用できます。電流入力が別の端子であることを除いて、DCIコマンドはDCVコマンドと同様に使用できます。また電流測定経路は、DC電圧および抵抗測定の場合の高速リード・リレーでなく、直列電機子リレーにより選択されます。このため、電流測定と他の機能との切り替えは、DCVと抵抗測定の切り替えよりも時間がかかります(30ms〜40ms)。

抵抗

抵抗測定は、DCV測定よりも長いセトリング時間を必要とします。10kΩ以上では、最初の測定が仕様リミット内で正しいことを保証するために、さらに長いセトリング時間を使用します。ここでも、最初の測定の確度を大目に見る場合は、高抵抗の測定のためのセトリング時間の代りに、デフォルトの遅延を使用できます。プログラムの遅延設定を小さく変更する前に、アプリケーションに対して最適なセトリング時間を決めるために、試行を行う必要があります。図45は、最初の読み取りを正確にするための、抵抗が増加したときのセトリング時間の増加の一般的な傾向を示しています。

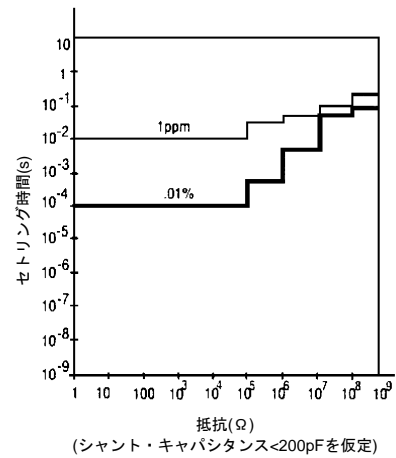
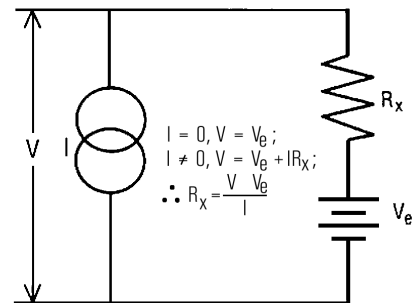
図45. 抵抗測定に対するセトリング時間の特性。被測定回路に200pF>のシャント・キャパシタンスを仮定。抵抗が小さいうちは、デフォルト値以下に遅延を設定する実用的なメリットはありません。抵抗が100kΩを超えると、最終値を得るには長いセトリング時間が必要です。したがって遅延時間を低く抑えると、測定確度の犠牲のうえで測定時間を抑えられる場合があります。

3458Aは、オフセット補正抵抗という機能も備えています。そのコンセプトはオートゼロに似ていて、まず熱起電力(DCVオフセット)を測るため、電流を印加せずに入力抵抗の測定を行います。そして、図46で示すように電流を印加し、未知抵抗の測定からオフセット電圧を減算して結果を表示します。オートゼロの場合と同様、未知抵抗の最終決定には2回の測定が行われます。しかし実際には、このようなオフセットは小さな抵抗値でのみ問題になります。3458Aは10mA電流源を備えており、これは少なくとも熱に起因するオフセット効果をマスクします。したがって小さな抵抗の測定では、オフセット補正抵抗は必要でない場合が多くあります。

図46. オフセット補正抵抗は、回路内の熱電対効果のような、小さな直列電圧源の影響を補正します。電流源をオフにして、未知抵抗の電圧Veを測定し、次に電流源をオンにして同様の測定を行います。これによって、Veの測定に対する影響が除かれます。

トラック・アンド・ホールド経路(直接サンプリングとサブサンプリング)の最適化

初めに説明したように、標準のDCV経路は信号を直接ADコンバータに送ります。この経路は帯域幅150kHzと選択可能な分解能4 1/2桁〜8 1/2桁を提供します。これに対して、トラック・アンド・ホールド経路の帯域幅は12MHz、分解能は4 1/2桁です。トラック・アンド・ホールド経路は入力とADコンバータ間にある16ビットのトラック・アンド・ホールド回路により、入力信号の「スナップショット」をとります。DVCは、最大読み取り速度50,000回/sの測定が可能です。この経路のためのコマンドを以下に示します。

DSAC (直接サンプリング、AC結合)
DSDC (直接サンプリング、DC結合)
SSAC (サブサンプリング、AC結合)
SSDC (サブサンプリング、DC結合)

これらのコマンドの詳しい使用法と、関連するトリガ・コマンド、制限については、Product Note 3458A-2「3458Aシステム・マルチメータを使用した高分解能ディジタイズ」で説明しています。一般に、これらのコマンドでスループットに最も影響を与えるのは、ACVに関係したものです。3458Aは、同期ACV法によって繰り返し波形の、またランダムACV法によってノイズ信号の実効値測定を行います。次のセクションでは、実効値ACVを測定する3つの方法について、それらのテクニックやトレードオフを説明します。

AC電圧とAC電流

3458Aマルチメータは、入力波形の等価DCV熱量(真の実効値)の測定のために、アナログACV、同期ACV、ランダムACVの3つの方法を提供します。入力信号はトラック・アンド・ホールド経路を進み(図47)、AC-DCコンバータ、またはトラック・アンド・ホールド回路へ導かれます。

アナログACV

アナログACV法はモノリシックAC-DCコンバータを使用して、広帯域(10Hz〜2MHz)の実効値機能を提供します。確度は高いものの、同期ACV法ほど良くはありません。また分解能も高いものの、ランダム法および同期ACV法ほど良くはありません。しかしその測定帯域幅にわたって、他の2つの方法よりも正確で高速な測定を提供します。また、繰り返し波形とノイズ信号を測定できます。

同期ACV

同期ACV法は、優れた100ppmのベスト確度とともに1Hz〜10MHzの帯域幅を提供しますが、入力は繰り返し波形である必要があります。読み取り速度は入力波形の周波数、および必要とする確度および分解能によって決まります。同期ACV法は直接的で、まず入力波形に対し周波数測定を行います。入力をシーケンシャルにサンプリングするか、20µs間隔のバーストでサンプリングするかは、周波数の値に基づいて決められます。そして、測定値が統計的に処理されて実効値が得られます。速度の尺度としてのサンプリング数は、選択した分解能によって決まり、測定の確度を決定します。

ランダムACV

ランダムACV法の測定帯域幅の上限は同期ACV法と同じですが、波形はノイズおよび非繰り返し信号が可能です。測定の分解能はサンプリング数に依存するため、高分解能では、この動作モードはACV機能のうちで確度および速度が最も低くなります。エリアシング(Product Note 3458A-2で詳述)は、サンプリング間隔のランダム選択(20〜40µs、10nsステップ)によって回避されています。

各ACVモードの比較

3つのACV動作モードのすべてについて、入力周波数が許す限り、ユーザは確度対速度のバランスを選択できます。表31を見ると、読み取り速度の周波数依存度はアナログACVで最もよく表れています(10Hz～1kHzで1回/s、1kHz～10kHzで10回/s、10kHz～2MHzで50回/s)。これらの読み取り速度は、アナログACVに対し仕様化された確度に関係しています。どの動作モードでも、低い分解能を選択、または遅延時間をデフォルトから入力信号の最大周波数成分の逆数の10倍に減らすことによって、測定速度を増加させることができます。例えば1kHzの信号を捕捉する場合に、その波形の代表的な測定のためには、最低で10msの遅延時間が必要です。

図47. ACV測定の3つの方法の信号経路のブロック図

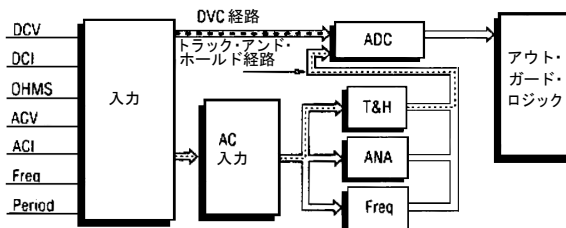



表31. ACVモードの比較

	アナログ	同期	ランダム
帯域幅	10Hz～2MHz	1Hz～10MHz	20Hz～10MHz
ベスト確度	300 ppm	100 ppm	1000 ppm
読み取り速度	50回/s	10回/s	40回/s
波高率	5:1	5:1	5:1
波形	全	繰り返し	全

AC電流

AC電流測定は、DCIシャントによって供給される電圧入力により、完全にアナログACVセクションで行われます。ACI測定のモードに関して実際的な判断基準はありませんが、積分およびセトリング時間を減らして、確度を犠牲にして読み取り速度を増加させることはできます。経験則から、代表的な実効値測定のために、AC-DCコンバータは入力波形の最低でも10周期を必要とします。そのため、入力信号の周波数は読み取り速度に直接影響します。ACVおよびACIをアプリケーションに適応させるため、測定スループットを微調整するには、3458Aの特性評価が必要になるでしょう。

周波数と周期

トラック・アンド・ホールド経路は、周波数およびその逆数である周期測定のために信号が通らなければならないルートでもあります。3458Aは、10Hz〜10MHz、7 1/2桁まで、最大ゲート時間1秒の周波数応答を提供します。確度や分解能ではなく速度が重要な場合は、内部カウンタのゲート時間を短くすることができます。表32は、各ゲート時間に対する分解能のトレードオフを示しています。

表32. 各ゲート時間に対する分解能の値

ゲート時間	分解能	読み取り速度
1秒	7 1/2桁	1回/秒
0.1秒	6 1/2桁	10回/秒
0.01秒	5 1/2桁	73回/秒
0.001秒	4 1/2桁	215回/秒
0.0001秒	3 1/2桁	270回/秒

十分な確度を伴った測定に対する最小時間という意味で、個々の測定を最適化した後でも、テスト・スループットを改善するもう一つの要素、タスクの割り当てがあります。この要素には、システム内のコントロール・コンピュータとその他の計測器が関係します。このプロダクト・ノートの導入部でも述べたように、一般的に、テスト・システム内の最速な装置はDMMです。したがって、その読み取り速度はシステムのスループットのためにボトルネックとはならないでしょう。3458Aの高速測定の利点を活かして、統計演算、サーミスタの線形化、リミットのチェックは3458Aに任せて、コントローラにはビジーでなければ他の装置のコントロールをさせます。内蔵の演算機能、読み取り値メモリ、ステート・メモリ、プログラム・メモリなどの3458A DMMの機能が、これを可能にします。

測定値やコマンドをコンピュータに転送するための時間は、コンピュータに依存します。GPIBのターンアラウンド時間(OUTPUTとENTER動作の処理時間)は、コンピュータによってかなり異なります。3458Aのプログラム・メモリ、読み取り値メモリ、ステート・メモリ、後処理演算の各機能は、すべてGPIBのオーバヘッドを減らす傾向にあり、テスト時間のコンピュータ依存度を大きく低減します。

タスク割り当てによるテスト・プロセスの最適化

演算動作

個々に見た場合、3458A内部の演算動作によって読み取り速度は遅くなります。しかし多くの場合、3458Aとコントローラを組み合わせて3458Aがいくらかの演算を行えば、最終結果は早く得られます。特に、テストがフェイルしたときだけコンピュータに警告する合否リミット・チェックの場合に、これが言えます。また測定のために統計が重要なときは、コンピュータ上でプログラムを書くよりも3458Aに計算作業をさせた方が高速です。3458Aが内蔵するコンピュータは、動作クロック8MHzのMotorola 68000です。したがって、3458Aの演算能力はコントローラに匹敵する場合も多くあります。

データ保存

測定データは、最大10,000個(20kB)までを3458Aのメモリに保存して、後でコントローラに転送できます。オプション001の拡張メモリを装備すると、さらに65,000個(128kB)の保存ができます。3458A読み取り値メモリへの転送速度、およびHP 9000 シリーズ200/300コンピュータでダイレクト・メモリ・アクセスを使用したGPIB転送速度は、1秒あたり100,000回です。この読み取り値メモリの利点は、いつでもコントローラにとって都合の良いときにデータにアクセスでき、また測定が終了するまでシステムを拘束する必要が

ないことです(高速のDMMでも、積分時間やセトリング時間が長い場合、また多くの測定をアベレージする場合には、システムを遅らせせることがあります)。

出力フォーマット

3458AはメモリおよびGPIB出力のために、単精度整数(SINT)、倍精度整数(DINT)、IEEE-728 4バイト単精度実数(SREAL)、IEEE-728 8バイト倍精度実数(DREAL)、およびASCIIの、5種類のデータ・フォーマットに対応しています。データ転送のために最も高速なのは、単精度整数型です。これは16ビットの整数形式で、小数点の配置のためにレンジ情報が既知でなければなりません。また16ビットしかないため、4 1/2桁より大きな桁が必要なときは他の形式を使う必要があります。次に高速なのは、倍精度整数型です。これは32ビットの整数フォーマットのため、レンジ情報を除いて全測定データを転送できます。SREALは4×8ビットの長さにより、レンジ情報を含む全データを転送します。これを使用するには、コントローラがこのフォーマットを受け付けてASCII形式に変換できなければなりません。最後に、最も遅いのがASCIIフォーマットです。この場合は、基本的に、測定値は18バイトのデータとキャリッジ・リターンおよび改行の終端子を持つ必要があります。多くの場合、データを高速に捕捉することが重要ですが、実際のデータ転送が低速な場合があります。この場合には、データ・フォーマットの最良の組み合わせとして、メモリへの入力にSREAL、GPIBへの出力にASCIIを使用します。HP 9000シリーズ200/300コンピュータは、変換の必要なくDINTおよびSINTを直接受け入れます。ASCIIフォーマットは、ほとんどのコントローラでも受け付けることができます。

測定ごとにファンクションやレンジが変わり、結果がコンピュータに保存されるプログラムでは、少し速度を落としてデータをDREALかSREALで読み取り値メモリに保存するのが最適です。これによって、SINTやDINTに必要なスケーリング・パラメータを常に記録しておく必要がなくなります。

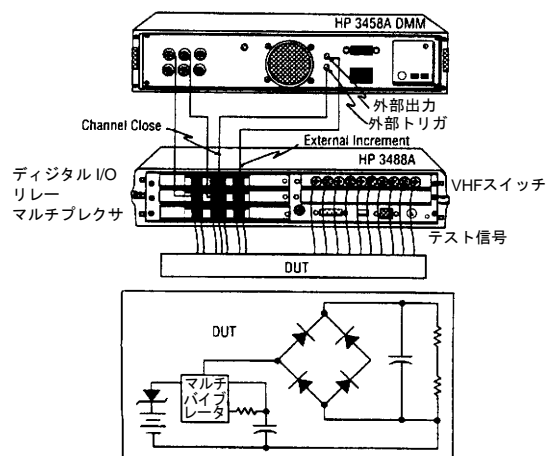
ステート保存とプログラム・メモリ

ステート保存およびプログラム・メモリの機能によって、テスト・タスクの適切なところで大きく時間を節減できます。ステート・メモリを使用して、GPIBから単一のコマンドを送って、測定器の静的ステートを確立します。初期化プログラムにより、システムのデッドタイムの間に、テスト・プログラムでプログラマーが使用したいステートをセットアップできます。その後、ステートは自由に呼びだせます。

プログラム・メモリは、動的メモリです。3458Aのステートは、イベントのシーケンスをコンピュータが制御しているかのように、プログラム・メモリ内でプログラムされた動作シーケンスが進むに連れて動的に変化します。測定データは、読み取り値メモリに保存し、都合の良いときに生データでコンピュータに転送したり、3458Aで後処理したりできます。ここでも、コマンド文字列をいったん3458Aのメモリに転送すれば、GPIBから簡単なコマンドを送るだけで測定シーケンスを開始できます。また、この簡単なコマンド送出による時間節減よりも重要なこととして、3458Aの構文解析ルーチンがプログラム・メモリのコマンド文字列をコンパイルするため、測定シーケンスはコンピュータが操作を制御するよりもはるかに速く実行されます。また、ステート・メモリや他のサブプログラムをプログラム・メモリから呼びだして、時間のかかるセットアップのプログラミング作業を効率化することもできます。

測定リスト

システム内での3458Aの最も効率的な使用法は、プログラム・メモリに測定リストを設定することです。この測定リストを、信号スイッチング装置のチャネル・リストに対応させます。3458Aの外部出力をスイッチング装置のChannel Advance入力に接続し、スイッチング装置のChannel Closed出力を3458Aの外部トリガ入力に接続します。チャネルのクローズや測定実行にどれだけの時間を要するかにかかわらず、チャネルは常にクローズされ、測定はWAITコマンドや遅延の付加なしに、常に完了のための時間を持つことになります。また、GPIBデータ・メッセージが減ることにより、より高速で効率的なプログラミングが可能になります。図48では、外部トリガ入力と外部出力を使用した、3458Aとスイッチング・ユニット3488Aの相互動作を示すための回路をテストしています。測定は、簡単なACおよびDC電圧および抵抗です。ここでは、チャネル閉が比較的に遅いため(3488Aは、スイッチング速度が約25ms/チャネル閉の、非常に汎用性の高い、しかし高速でない電機子リレーを使用しています)、ファンクションやレンジを変更するための時間がテスト・セットアップに重要です。したがって、1つのテスト・ポイントで複数の測定を行なっています。もしリード・リレーを使用した場合は、一般的に、テスト状況が許す限り、同じファンクションのままでのテスト・ポイントの変更の方がより高速となります。

図48. External Incrementを外部出力と、Channel Closeを外部トリガと接続して、測定リストとスキャン・リストによりテスト・スループットを増大

ベンチマーク

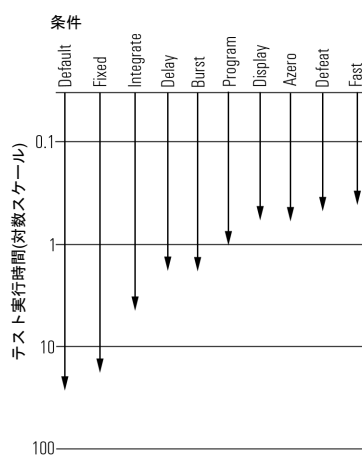
3458Aマルチメータのさまざまなファンクションの影響を示すためのベンチマークは、最も便利ですが最も低速なプロシージャから開始します。このプロシージャでは、コンピュータがDMMにファンクションの変更を要求し、測定を実行し、そして測定結果をコンピュータに転送します。このベンチマークでは、無限に高速のスイッチング速度と無限のダイナミック・レンジを持ったFETスキャナの使用を仮定しています。したがって、このベンチマークは仮想的な状況を表しますが、3458Aのさまざまな動作モードの違いを最も良く示すものとなっています。コンピュータは、HP 9000シリーズ200/300を使用します。他コンピュータの場合の時間は、そのGPIBターンアラウンド時間によって異なります。ベンチマークの結果は図49で示しています。

DUTの構成には、以下が含まれています。

25個の抵抗測定
15 <10kΩ ±5%
8 <100kΩ ±5%
2 <10kΩ ±0.001%

10個のDCV測定
5 <30V ±1%
4 <10V ±0.01%
1 <1V ±0.001%

3個のACV測定
1 <250V @50Hz @±5%
1 <10V @25kHz @±0.1%
1 <1V @5kHz @±0.075%

図49. さまざまな構成でのベンチマーク実行時間

測定シーケンスは、回路をオンにする前に抵抗値を調べることを要求します。次に、電源ライン電圧が適正レベルかをチェックします。出力レベル1V @5kHzを、±.075%のリミットでチェックします。最後に、残りの電圧を次の順序でチェックします。

```
2 DCV <10 V ±1%
1 DCV <10 V ±.01%
2 DCV <10 V ±1%
```

```
1 DCV <1 V ±.001%
1 ACV <10 V ±.1%
1 DCV <10 V ±1%
3 DCV <10 V ±.01%
```

ベンチマーク結果

デフォルト条件: (サブプログラムDefault) 時間 = 20.63秒

```
560 SUB Default(REAL Dnld_time.Exe_time,Tns_time)
570 DIM A(37)
580 Exe_time=TIMEDATE
590 OUTPUT 722;"RESET;TRIG SYN"
600 OUTPUT 722;"OHM"
610 FOR I=1 TO 23
620 ENTER 722;A(l)
630 NEXT I
640 OUTPUT 722;"OHMF"
...
780 ENTER 722;A(I)
790 NEXT I
800 Exe_time=TlMEDATE-Exe_time
810 Dnld_time=O
820 Tns_time=0
830 SUBEND
```

コンピュータは3458Aをリモート動作にし、またデフォルト状態にリセットします。積分時間を10PLCに設定し、セトリング遅延は、ファンクションまたはレンジ変更後の最初の測定がその仕様確度内となるように設定します。オートレンジはオンです。コンピュータは37個の測定のうちの10個の測定について、レンジ、またはファンクション、または積分時間の変更をDMMに要求します。その他は同じ測定設定により、ブロックで測定します。

固定レンジ: (サブプログラムFixed) 時間 = 15.98秒

```
840 SUB Fixed(REAL Dnld_time,Exe_time,Tns_time)
850 DIM A(37)
860 Exe_time=TIMEDATE
870 OUTPUT 722;"RESET;TRIG SYN"
880 OUTPUT 722;"OHM,lE4"
890 FOR I=1 TO 15
900 ENTER 722;A(I)
910 NEXT I
920 OUTPUT 722;"OHM,1E5
...
1110 ENTER 722:A(I)
1120 NEXTI
1130 Exe_time=TIMEDATE-Exe_time
1140 Dnld_time=0
1150 Tns_time=0
1160 SUBEND
```

オートレンジの代りに、レンジを測定に必要なレンジに固定することを除いて、テスト条件はデフォルトと同じです。

適切な積分時間: (サブプログラムIntegrate) 時間 = 3.76秒

```
1170 SUB Integrat(REAL Dnld_time,Exe_time,Tns_time)
1180 DIM A(37)
1190 Exe_time=TIMEDATE
1200 OUTPUT 722;"PRESET"
1210 OUTPUT 722;"OHM,lE4;NPLC 0"
1220 FOR I=1 TO 15
1230 ENTER 722;A(I)
1240 NEXT I
1250 OUTPUT 722;"OHM,1E5"
...
1410 ENTER 722;A(34)
1420 OUTPUT 722;"DCV,10;NPLC 0"
1430 FOR I=35 TO 37
1440 ENTER 722;A(I)
1450 NEXT I
1460 Exe_time=TIMEDATE-Exe_time
1470 Dnld_time=0
1480 Tns_time=0
1490 SUBEND
```

積分時間がデフォルトの10PLCでなく、各測定に必要な分解能と確度のために適切な時間であることを除いて、テスト条件は「固定レンジ」の条件と同じです。

適切な遅延時間: (サブプログラムDelay) 時間 = 1.48秒

```
1500 SUB Delay(REAL Dnld_time,Exe_time,Tns_time)
1510 DIM A(37)
1520 Exe_time=TIMEDATE
1530 OUTPUT 722;"PRESET"
1540 OUTPUT 722;"OHM,1E4;NPLC 0;DELAY 0
1550 FOR I=1 TO 15
...
1730 OUTPUT 722;"ACV,10;ACBAND 5000;APER 20E-6;DELAY .01"
1740 ENTER 722;A(34)
1750 OUTPUT 722;"DCV,10;NPLC 0;DELAY 0"
1760 FOR I=35 TO 37
1770 ENTER 722;A(I)
1780 NEXT I
1790 Exe_time=TIMEDATE-Exe_time
1800 Dnld_time=0
1810 Tns_time=0
1820 SUBEND
```

遅延時間がデフォルトでなく、各測定に必要な確度のために適切な時間であることを除いて、テスト条件は「適切な積分時間」の条件と同じです。

読み取り値メモリの使用: (サブプログラムBurst)

テスト実行時間 = 1.42秒

読み取り値値転送時間 = 0.18秒

```
1830 SUB Burst(REAL Dnld_time,Exe_time,Tns_time)
1840 DIM A(37)
1850 Exe_time=TIMEDATE
1860 OUTPUT 722;"PRESET;MEM FIFO;MFORMAT SREAL
1870 OUTPUT 722;"OHM,1E4;NPLC 0;DELAY 0;NRDGS 15;TRIG SGL
1880 OUTPUT 722;"OHM,1E5;NRDGS 8;TRIG SGL
```

```
...
1940 OUTPUT 722;"DCV,10;NPLC 0;DELAY 0;NRDGS 3;TRIG SGL
1950 Exe_time=TIMEDATE-Exe_time
1960 Dnld_time=0
1970 Tns_time=TIMEDATE
1980 FOR I=1 TO 37
1990 ENTER 722;A(I)
2000 NEXT I
2010 Tns_time=TlMEDATE-Tns_time
2020 SUBEND
```

プログラムの構造に際立った変更がなされています。読み取り値は、測定が実行されると読み取り値メモリに保存されます。測定シーケンスの終りに、測定値はFOR NEXTループにより読み取り値メモリからコンピュータに転送されます。データ転送の他には、読み取り速度の面で際立った改善はありません。コンピュータへのデータ転送をTRANSFER文により行えば、さらに時間短縮が可能です。

プログラム・メモリの使用: (サブプログラムProgram)
テスト実行時間 = 1.06秒
プログラム・メモリ・ダウンロード時間 = 0.260秒
読み取り値転送時間 = 0.17秒

```
2030 SUB Program(REAL Dnld_time,Exe_time,Tns_time)
2040 DIM A(37)
2050 Dnld_time=TlMEDATE
2060 OUTPUT 722;"PRESET;MFORMAT SREAL"
2070 OUTPUT 722;"SUB 1;MEM FIFO;OHM,lE4;NPLC 0;DELAY0;NRDGS 15;TRIG SGL"
2080 OUTPUT 722;"OHM,1E5;NRDGS 8:TRIG SGL"
2090 OUTPUT 722;"OHMF,1E3;APER 20E-6;DELAY-l;NRDGS 2;TRIG SGL
2100 OUTPUT 722;"ACV,250;ACBAND 250;DELAY.1;NRDGS 1;TRIG SGL"
2110 OUTPUT 722;"ACV 10;ACBAND 25000;DELAY .01;TRIG SGL
2120 OUTPUT 722;"DCV,10;NPLC 0;DELAY 0;NRDGS 6;TRIG SGL"
2130 OUTPUT 722:"ACV,10:ACBAND 5000;APER 20E-6;DELAY.01;NRDGS 1;TRIG SGL
2140 OUTPUT 722;"DCV,10;NPLC 0;DELAY 0;NRDGS 3;TRIG SGL;SUBEND"
2150 Dnld_time=TlMEDATE-Dnld_time
2160 Exe_time=TIMEDATE
2170 OUTPUT 722;"CALL 1"
2180 Exe_time=TIMEDATE-Exe_time
2190 Tns_time=TIMEDATE
2200 FOR I=1 TO 37 2210ENTER 722:A(I)
2220 NEXT I
2230 Tns_time=TIMEDATE-Tns_time
2240 SUBEND
...
```

これも、プログラムの構造を変更しています。これまでに加えられたすべての変更とともに、測定シーケンスは、DMMサブプログラムSUB 1に置かれます。コマンドはコンピュータから3458Aに転送されて、3458Aがそれをコンパイルします。コマンドの実行は、CALL 1コマンドによるDMMサブプログラムの呼び出しとともに開始します。DMMサブプログラムが完了すると、測定値が転送されます。

DMMがこのシーケンスを実行する間、プログラムは停止します。もしプログラムを継続させる場合は、DMMへの出力文を

```
OUTPUT 722 USING "#,K"; "CALL 1"
```

にします。イメージ"#,K"の使用により、終端子EOL(End-Of-Line)が抑制されます。3458Aは終端子を持たないコマンドを受け取るとコンピュータを解放し、コンピュータはプログラムを継続できます。その間に3458Aは、要求された動作を続けます。ここでは、ベンチマークの実行時間は、単に読み取り値メモリを使うだけの場合よりも際立って短縮されています。

ディスプレイのオフ: (サブプログラムDisplay)
テスト実行時間 = 0.500秒
プログラム・メモリ・ダウンロード時間 = 0.280秒
読み取り値転送時間 = 0.180秒

```
2250 SUB Disp(REAL Dnld_time,Exe_time,Tns_time)
2260 DIM A(37)
2270 Dnld_time=TIMEDATE
2280 OUTPUT 722;"PRESET;MFORMAT SREAL;DISP OFF,TESTING"
2290 OUTPUT 722;"SUB 1;MEM FIFO;OHM,1E4;NPLC 0;DELAY 0;NRDGS 15;TRIG SGL"
2300 OUTPUT 722;"OHM,1E5;NRDGS 8;TRIG SGL"
2310 OUTPUT 722;"OHMF,1E3;APER 20E-6;DELAY -1 ;NRDGS 2;TRIG SGL"
...
```

このプログラムは、ディスプレイをオフにすることを除いて、サブプログラムProgramと同じです。テスト実行時間は半分に短縮されています。

オードゼロのオフ: (サブプログラムAzero)
テスト実行時間 = 0.510秒
プログラム・メモリ・ダウンロード時間 = 0.280秒
読み取り値転送時間 = 0.180秒

このプログラムは、オートゼロをオフにすることを除いて、サブプログラムDisplayと同じです。読み取り速度が非常に高速なため、オートゼロのオン／オフに実質的な違いがないため、この方法による実質的なメリットはありません。場合によっては、ファンクションや積分時間を変更する場合は、オートゼロ・オンのほうが高速なことがあります。

無効機能のオン: (サブプログラムDefeat)
テスト実行時間 = 0.470秒
プログラム・メモリ・ダウンロード時間 = 0.280秒
読み取り値転送時間 = 0.180秒

```
2690 SUB Defeat(REAL Dnld_time,Exe_time,Tns_time)
2700 DIM A(37)
2710 Dnld_time=TIMEDATE
2720 OUTPUT 722;"PRESET;DISP OFF,TESTING;MFORMAT SREAL;DEFEAT ON"
2730 OUTPUT 722;"SUB 1;MEM FIFO;OHM,1E4;NPLC 0;DELAY 0;NRDGS 15;TRIG SGL"
2740 OUTPUT 722;"OHM,1E5;NRDGS 8;TRIG SGL"
```

このプログラムは、DEFEAT(無効)機能をオンにすることを除いて、サブプログラムDisplayと同じです。この動作モードでは、過負荷の検出・保護回路の一部を無効にします。300Vの電圧を検出するとこの無効機能はオフとなり、イベントが3458Aのメモリに記録されます。この機能はファンクションおよびレンジの変更を高速にしますが、乱用は避けるべきです。

さらなる高速化

OUTPUTおよびENTER文の代りにTRANSFERを使用すると、スループットがかなり向上します。また、いくつかのコマンドはその並立化によって、読み取り速度が改善されます。特にACVにおけるDELAYとACBANDの順序は、実行速度にかなりの違いをもたらします。適切な順序を示します。

```
DELAY <#>;ACBAND <#,#>;ACV <range>.
```

ファンクションの変更でデフォルトのセトリング時間を変更する場合は、常に最初にDELAY文を変更します。また積分時間は、変更するよりも一つを維持したほうが高速なことが多くあります。例えば、6 1/2桁の分解能を得るのにAPER 10E-5(100µs)と設定でき、これは1秒間に約10,000測定を可能にします。この分解能で数回の測定しか呼び出さず、より低い分解能で多くの測定を呼び出す場合でも、100µsの積分時間を維持して全測定を行なったほうが高速なことがあります。積分時間を変更するのに、3458Aは6ms〜10msの時間がかかります。1秒あたり約10,000回の測定の場合、3458Aはその時間に100回の6 1/2桁測定を行えます。

最後のプログラムは、転送と適切なコマンド・シーケンスを使用して、ベンチマーク・プログラムに対する最大限のスループットを実現しています。

実行時間 = 0.360秒

プログラム・メモリ・ダウンロード時間 = 0.05秒

測定値転送時間 = 0.05秒

```
 10 OPTION BASE 1
 20 DIM Command$[1000] BUFFER
 30 DIMA$[100],B$[100],C$[100],D$[100],E$(100],
    F$[100],G$[100],H$[100],I$[100],Set_up$[100]
 40 INTEGER I,M
 50 REAL Readings(37) BUFFER
 60 ASSIGN @Dmm TO 722
 70 ASSIGN @Buf_1 TO BUFFER Command$
 80 ASSIGN @Buf_2 TO BUFFER Readings(*)
 90 CLEAR 722
100 OUTPUT @Dmm;"RESET"
110 Set_up$="PRESET;MFORMAT SREAL;DEFEAT ON;APER 100E-6;DISP OFF,TESTING"
120 B$="SUB Try;MEM FIFO;DELAY O;OHM,1E4;NRDGS 15;TRIG SGL;"
130 C$="OHM,1E5;NRDGS 8;TRIG SGL;"
140 D$="DELAY -1;OHMF,1E3;NRDGS 2;TRIG SGL;"
150 E$="DELAY .1;ACBAND 5O;ACV 250;NRDGS 1;TRIG SGL;
160 F$="DELAY .01;ACBAND 25000;ACV,10;TRIG SGL;"
170 G$="DELAY 0;DCV 10;NRDGS 6;TRIG SGL;"
180 H$="DELAY .01;ACBAND 5000;ACV 10;NRDGS 1; TRIG SGL;"
190 I$="DELAY 0;DCV 10;NRDGS 3;TRIG SGL;SUBEND"
200 Command$=B$&C$&D$&E$&F$&G$&H$&I$
210 Dnload:! DMMへの転送コマンド
220 Dnld_time=TIMEDATE
230 OUTPUT @Dmm;Set_up$
240 TRANSFER @Buf_1 TO @Dmm
250 Dnld_time=TIMEDATE-Dnld_time
260 Execute: ! DMM実行時間
270 Exe_time=TIMEDATE
280 OUTPUT @Dmm;"CALL Try"
290 Exe_time=TIMEDATE-Exe_time
300 Read:! 読み取り値をコンピュータに転送
310 Tns_time=TIMEDATE
320 TRANSFER @Dmm TO @Buf_2
330 Tns_time=TIMEDATE-Tns_time
340 PRINT "DOWN LOAD TIME =";Dnld_time
```

```
350 PRINT "EXECUTION TIME =";Exe_time
360 PRINT "TRANSFER TIME = ";Tns_time
370 PRINT "TOTAL TIME = "; Dnld_time+Exe_time+Tns_time
380 END
 10 ! ベンチマーク・テスト
 20 !
 30 COM Dnld_trme.Exe_time,Tns_time
 40 !
 50 CALL Default(Dnld_time,Exe_time,Tns_time)
 60 PRINT USING "36A.DD.DDD";"The execution time for default is
    ";Exe_time
 70 PRINT
 80 !
 90 CALL Fixed(Dnld_time.Exe_time,Tns_time)
100 PRINT USING "38A.DD.DDD":"The execution time for fixed range is
    ";Exe_time
110 PRINT
120 !
130 CALL Integrat(Dnld_time,Exe_time,Tns_time)
140 PRINT USING "51A,DD.DDD";"The execution time for correct integration
    time is";Exe_time
150 PRINT
160 !
170 CALL Delay(Dnld_time,Exe_time,Tns_time)
180 PRINT USING "44A,DD.DDD";"The execution time for correct delay time
    is";Exe_time
190 PRINT
200 !
210 CALL Burst(Dnld_time,Exe_time,Tns_time)
220 PRINT USING "44A,DD.DDD";"The execution time for storing readings
    is";Exe_time
230 PRINT USING "44A,DD.DDD";"The transfer time using FOR NEXT is
    ";Tns_time
240 PRINT USING "44A,DD,DDD";"The total time for memory IS
    ";Exe_time+Tns_time
250 PRINT
260 !
270 CALL Program(Dnld_time,Exe_time,Tns_time)
280 PRINT USING "44A,DD.DDD";"The execution time for program memory is
    ";Exe_time
290 PRINT USING "44A,DD,DDD";"The download time for transfering the SUB
    is";Dnld_time
300 PRINT USING "44A,DD.DDD";"The transfer time using FOR NEXT is
    ";Tns_time
310 PRINT USING "44A,DD.DDD";"The total time for program memory
    is";Exe_time+Dnld_time+ Tns_time
320 PRINT
330 !
340 CALL Disp(Dnld_time,Exe_time,Tns_time)
350 PRINT USING "44A,DD.DDD";"The execution time for program memory is
    ";Exe_time
360 PRINT USING "44A,DD.DDD";"The download time for transfering the SUB
    is";Dnld_time
370 PRINT USING "44A,DD.DDD";"The transfer time using FOR NEXT is
    ";Tns_time
380 PRINT USING "44A,DD.DDD";"The total time for display off
    is";Exe_time+Dnld_time+Tns_time
390 PRINT
400 !
410 CALL Azero(Dnld_time,Exe_time,Tns_time)
420 PRINT USING "44A,DD.DDD";"The execution time for program memory is
    ";Exe_time
430 PRINT USING "44A,DD.DDD";"The download time for transfering the SUB
    is";Dnld_time
```

```
 440 PRINT USING "44A,DD,DDD";"The transfer time using FOR NEXT is
     ";Tns_time
 450 PRINT USING "44A,DD.DDD";"The total time for AZERO off is";
     Exe_time+Dnld_time+ Tns_time
 460 PRINT
 470 !
 480 CALL Defeat(Dnld_time,Exe_time,Tns_time)
 490 PRINT USING "44A,DD.DDD";"The execution time for program memory is
     ";Exe_time
 500 PRINT USING "44A,DD,DDD";"The download time for transfering the SUB
     is";Dnld_time
 510 PRINT USING "44A,DD,DDD";"The transfer time using FOR NEXT is
     ";Tns_time
 520 PRINT USING "44A,DD.DDD";"The total time for DEFEAT ON
     is";Exe_time+Dnld_time+Tns_time
 530 PRINT
 540 !
 550 END
 560 SUB Default(REAL Dnld_time,Exe_time, Tns_time)
 570 DIM A(37)
 580 Exe_time=TlMEDATE
 590 OUTPUT 722:"RESET;TRIG SYN"
 600 OUTPUT 722;"OHM"
 610 FOR I=1 TO 23
 620 ENTER 722;A(I)
 630 NEXT I
 640 OUTPUT 722;"OHMF"
 650 ENTER 722;A(24)
 660 ENTER 722;A(25)
 670 OUTPUT 722;"ACV"
 680 ENTER 722;A(26)
 690 ENTER 722;A(27)
 700 OUTPUT 722;"DCV"
 710 FOR I=28 TO 33
 720 ENTER 722;A(I)
 730 NEXT I
 740 OUTPUT 722;"ACV"
 750 ENTER 722;A(34)
 760 OUTPUT 722;"DCV"
 770 FOR I=35 TO 37
 780 ENTER 722;A(I)
 790 NEXT I
 800 Exe_time=TIMEDATE-Exe_time
 810 Dnld time=0
 820 Tns_time=0
 830 SUBEND
 840 SUB Fixed(REAL Dnld_time,Exe_time,Tns_time)
 850 DIM A(37)
 860 Exe_time=TIMEDATE
 870 OUTPUT 722;"RESET;TRIG SYN"
 880 OUTPUT 722;"OHM,1E4
 890 FOR I=1 TO 15
 900 ENTER 722;A(I)
 910 NEXT I
 920 OUTPUT 722;"OHM,1E5"
 930 FOR I=16 TO 23
 940 ENTER 722;A(I)
 950 NEXT I
 960 OUTPUT 722;"OHMF,1E3"
 970 ENTER 722;A(24)
 980 ENTER 722;A(25)
 990 OUTPUT 722;"ACV,250;ACBAND 250"
1000 ENTER 722;A(26)
1010 OUTPUT 722;"ACV 10;ACBAND 25000"
1020 ENTER 722;A(27)
```

```
1030 OUTPUT 722;"DCV, 10"
1040 FOR I=28 TO 33
1050 ENTER 722;A(l)
1060 NEXT I
1070 OUTPUT 722;"ACV,10;ACBAND 5000"
1080 ENTER 722;A(34)
1090 OUTPUT 722;"DCV,10"
1100 FOR I=35 TO 37
1110 ENTER 722;A(I)
1120 NEXT I
1130 Exe_time=TIMEDATE-Exe_time
1140 Dnld time=0
1150 Tns_time=0
1160 SUBEND
1170 SUB Integrat(REAL Dnld_time,Exe_time, Tns_time)
1180 DIM A(37)
1190 Exe_time=TIMEDATE
1200 OUTPUT 722;"PRESET"
1210 OUTPUT 722;"OHM,1E4;NPLC 0"
1220 FOR I=1 TO 15
1230 ENTER 722;A(I)
1240 NEXT I
1250 OUTPUT 722;"OHM,1E5"
1260 FOR I=16 TO 23
1270 ENTER 722;A(l)
1280 NEXT I
1290 OUTPUT 722;"OHMF,1E3;APER 20E-6"
1300 ENTER 722;A(24)
1310 ENTER 722;A(25)
1320 OUTPUT 722;"ACV,250;ACBAND 250"
1330 ENTER 722;A(26)
1340 OUTPUT 722;"ACV 10;ACBAND 25000"
1350 ENTER 722;A(27)
1360 OUTPUT 722;"DCV,10;NPLC 0"
1370 FOR I=28 TO 33
1380 ENTER 722;A(I)
1390 NEXT I
1400 OUTPUT 722:"ACV,10;ACBAND 5000;APER 20E-6"
1410 ENTER 722;A(34)
1420 OUTPUT 722;"DCV,10;NPLC 0"
1430 FOR I=35 TO 37
1440 ENTER 722;A(I)
1450 NEXT I
1460 Exe_time=TIMEDATE-Exe_time
1470 Dnld_time=0
1480 Tns_time=0
1490 SUBEND
1500 SUB Delay(REAL Dnld_time,Exe_time,Tns_time)
1510 DIM A(37)
1520 Exe_time=TIMEDATE
1530 OUTPUT 722;"PRESET"
1540 OUTPUT 722;"OHM,1E4;NPLC 0;DELAY 0"
1550 FOR I=1 TO 15
1560 ENTER 722;A(I)
1570 NEXT I
1580 OUTPUT 722;"OHM,1E5"
1590 FOR I=16 TO 23
1600 ENTER 722;A(I)
1610 NEXT I
1620 OUTPUT 722;"OHMF,1E3;APER 20E-6;DELAY-1"
1630 ENTER 722;A(24)
1640 ENTER 722;A(25)
1650 OUTPUT 722;"ACV,250;ACBAND 250;DELAY.1"
1660 ENTER 722;A(26)
1670 OUTPUT 722;"ACV 10;ACBAND 25000;DELAY.01"
```

```
1680 ENTER 722;A(27)
1690 OUTPUT 722;"DCV,10;NPLC 0;DELAY 0"
1700 FOR I=28 TO 33
1710 ENTER 722;A(I)
1720 NEXT I
1730 OUTPUT 722;"ACV,10;ACBAND 5000;APER 20E-6; DELAY .01"
1740 ENTER 722;A(34)
1750 OUTPUT 722;"DCV,10;NPLC 0;DELAY 0"
1760 FOR I=35 TO 37
1770 ENTER 722;A(I)
1780 NEXT I
1790 Exe_tlme=TIMEDATE-Exe_time
1800 Dnld_time=0
1810 Tns_time=0
1820 SUBEND
1830 SUB Burst(REAL Dnld_time,Exe_time,Tns_time)
1840 DIM A(37)
1850 Exe_time=TIMEDATE
1860 OUTPUT 722;"PRESET;MEM FIFO;MFORMAT SREAL"
1870 OUTPUT 722;"OHM,1E4;NPLC 0;DELAY 0;NRDGS 15:TRIG SGL"
1880 OUTPUT 722:"OHM,1E5;NRDGS 8;TRIG SGL"
1890 OUTPUT 722;"OHMF,1ES;APER 20E-6;DELAY -1;NRDGS 2,TRIG SGL"
1900 OUTPUT 722;"ACV,250;ACBAND 250:DELAY.1;NRDGS 1;TRIG SGL"
1910 OUTPUT 722;"ACV 10;ACBAND 25000;DELAY.01;TRIG SGL"
1920 OUTPUT 722;"DCV,10;NPLC 0;DELAY 0;NRDGS 6;TRIG SGL"
1930 OUTPUT 722;"ACV,10;ACBAND 5000;APER 20E-6;DELAY .01;NRDGS 1;TRIG SGL"
1940 OUTPUT 722;"DCV,10;NPLC 0;DELAY 0;NRDGS 3;TRIG SGL"
1950 Exe_time=TIMEDATE-Exe_time
1960 Dnld_time=0
1970 Tns_time=TIMEDATE
1980 FOR I=1 TO 37
1990 ENTER 722:A(I)
2000 NEXT I
2010 Tns_time=TIMEDATE-Tns_time
2020 SUBEND
2030 SUB Program(REAL Dnld_time,Exe_time,Tns_time)
2040 DIM A(37)
2050 Dnld_time=TIMEDATE
2060 OUTPUT 722:"PRESET;MFORMAT SREAL"
2070 OUTPUT 722;"SUB 1:MEM FIFO;OHM.1E4;NPLC 0;DELAY 0:NRDGS 15;TRIG SGL"
2080 OUTPUT 722;"OHM.1E5;NRDGS 8;TRIG SGL"
2090 OUTPUT 722,"OHMF, 1E3;APER 20E-6:DELAY-1; NRDGS 2;TRIG SGL"
2100 OUTPUT 722;"ACV,250;ACBAND 250;DELAY.1;NRDGS 1; TRIG SGL"
2110 OUTPUT 722;"ACV10;ACBAND 25000;DELAY.01;TRIG SGL"
2120 OUTPUT 722:"DCV,10;NPLC 0;DELAY 0;NRDGS 6; TRIG SGL"
2130 OUTPUT 722;"ACV,10;ACBAND 5000;APER 20E-6; DELAY.01;NRDGS 1;TRlG SGL"
2140 OUTPUT 722;"DCV,10;NPLC 0;DELAY 0;NRDGS 3; TRIG SGL;SUBEND"
2150 Dnld_time=TIMEDATE-Dnld_time
2160 Exe_time=TIMEDATE
2170 OUTPUT 722;"CALL 1"
2180 Exe_time=TIMEDATE-Exe_time
2190 Tns_time=TIMEDATE
2200 FOR I=1 TO 37
2210 ENTER 722;A(I)
2220 NEXT I
2230 Tns_time=TIMEDATE-Tns_time
2240 SUBEND
2250 SUB Disp(REAL Dnld_time,Exe_time,Tns_time)
2260 DIM A(37)
2270 Dnld_time=TIMEDATE
2280 OUTPUT 722;"PRESET;MFORMAT SREAL;DISP OFF,TESTING"
2290 OUTPUT 722;"SUB 1;MEM FIFO;OHM,1E4;NPLC 0;DELAY 0;NRDGS 15;TRIG SGL"
2300 OUTPUT 722;"OHM,1E5;NRDGS 8;TRIG SGL"
2310 OUTPUT 722;"OHMF,1E3;APER 20E-6;DELAY -1;NRDGS 2;TRIG SGL"
2320 OUTPUT 722;"ACV,250;ACBAND 250;DELAY .1;NRDGS 1:TRIG SGL"
```

```
2330 OUTPUT 722;"ACV 10;ACBAND 25000;DELAY .01;TRIG SGL"
2340 OUTPUT 722:"DCV,10;NPLC 0;DELAY 0;NRDGS 6;TRIG SGL"
2350 OUTPUT 722;"ACV,10;ACBAND 5000;APER 20E-6;DELAY .01;NRDGS 1;TRIG SGL"
2360 OUTPUT 722:"DCV,10;NPLC 0;DELAY 0;NRDGS 3;TRIG SGL;SUBEND"
2370 Dnld_time=TIMEDATE-Dnld_time
2380 Exe_time=TIMEDATE
2390 OUTPUT 722;"CALL 1"
2400 Exe_time=TIMEDATE-Exe_time
2410 Tns_time=TIMEDATE
2420 FOR I=1 TO 37
2430 ENTER 722:A(I)
2440 NEXT I
2450 Tns_time=TIMEDATE-Tns_time
2460 SUBEND
2470 SUB Azero(REAL Dnld_time,Exe_time,Tns_time)
2480 DIM A(37)
2490 Dnld_time=TIMEDATE
2500 OUTPUT 722;"PRESET;MFORMAT SREAL;DISP OFF, TESTING;AZERO OFF"
2510 OUTPUT 722;"SUB 1;MEM FIFO;OHM,1E4;NPLC 0;DELAY 0;NRDGS 15;TRIG SGL"
2520 OUTPUT 722;"OHM,1E5;NRDGS 8;TRIG SGL"
2530 OUTPUT 722;"OHMF,1E3;APER 20E-6;DELAY-1; NRDGS 2;TRIG SGL"
2540 OUTPUT 722;"ACV,250:ACBAND 250;DELAY.1;NRDGS 1;TRIG SGL"
2550 OUTPUT 722;"ACV 10;ACBAND 25000;DELAY.01; TRIG SGL"
2560 OUTPUT 722;"DCV10;NPLC 0;DELAY 0;NRDGS 6; TRIG SGL"
2570 OUTPUT 722;"ACV,10;ACBAND 5000;APER 20E-6; DELAY.01;NRDGS 1;TRlG SGL"
2580 OUTPUT 722;"DCV,10;NPLC 0;DELAY 0;NRDGS 3; TRIG SGL;SUBEND"
2590 Dnld_time=TIMEDATE-Dnld_time
2600 Exe_time=TIMEDATE
2610 OUTPUT 722;"CALL 1"
2620 Exe_time=TIMEDATE-Exe_time
2630 Tns_tlme=TIMEDATE
2640 FOR I=1 TO 37
2650 ENTER 722;A(I)
2660 NEXT I
2670 Tns_time=TIMEDATE-Tns_time
2680 SUBEND
2690 Defeat(REAL Dnld_time,Exe_time,Tns_time)
2700 DIM A(37)
2710 Dnld_time=TIMEDATE
2720 OUTPUT 722;"PRESET;DISP OFF,TESTING;MFORMAT SREAL;DEFEAT ON"
2730 OUTPUT 722;"SUB 1;MEM FIFO;OHM.1E4;NPLC 0;DELAY 0:NRDGS 15;TRIG SGL"
2740 OUTPUT 722:"OHM.1E5:NRDGS 8;TRIG SGL"
2750 OUTPUT 722;"OHMF,1E3;APER 20E-6:DELAY -1; NRDGS 2:TRIG SGL"
2760 OUTPUT 722;"ACV,250;ACBAND 250;DELAY .1; NRDGS 1;TRIG SGL"
2770 OUTPUT 722;"ACV 10;ACBAND 25000;DELAY.01; TRIG SGL"
2780 OUTPUT 722;"DCV,10;NPLC 0;DELAY 0;NRDGS 6;TRIG SGL"
2790 OUTPUT 722;"ACV,10;ACBAND 5000;APER 20E-6;DELAY .01;NRDGS 1;TRIG SGL"
2800 OUTPUT 722;"DCV10;NPLC 0;DELAY 0;NRDGS 3;TRIG SGL;SUBEND"
2810 Dnld_time=TIMEDATE-Dnld_time
2820 Exe_time=TIMEDATE
2830 OUTPUT 722;"CALL 1"
2840 Exe_time=TIMEDATE-Exe_time
2850 Tns_time=TIMEDATE
2860 FOR I=1 TO 37
2870 ENTER 722;A(I)
2880 NEXT I
2890 Tns_time=TlMEDATE-Tns_time
2900 SUBEND
```

```
  10 !メイン・プログラム
  20 COM A(20),B(90),C(30),D(30),J$[80]
  30 CALL Test_58(Time58)
  40 END
  50 !
  60 !
```

```
 70 SUB Test_58(Time58)
 80 DIM A(20),B(90),C(30),D(30),J$[80]
 90 !スキャナのセットアップ
100 ASSIGN @Scan TO 709
110 ASSIGN @Dmm TO 722
120 CLEAR @Dmm
130 OUTPUT @Dmm;"RESET"     !DMMを電源投入時の状態に設定
140 OUTPUT @Dmm;"TRIG HOLD" ! トリガを停止
150 !
160 ! -------- スキャナのセットアップ --------
170 !
180 OUTPUT @Scan;"RESET"
190 OUTPUT @Scan;"CLOSE 200,400,410;STORE 1"
200 OUTPUT @Scan;"CLOSE 308,309;STORE 2"
210 OUTPUT @Scan;"OPEN 200"
220 OUTPUT @Scan;"CLOSE 201; STORE 3"
230 OUTPUT @Scan;"OPEN 201"
240 OUTPUT @Scan;"CLOSE 206;STORE 4"
250 OUTPUT @Scan;"OPEN 206"
260 OUTPUT @Scan;"CLOSE 202;STORE 5"
270 OUTPUT @Scan;"OPEN 202"
280 OUTPUT @Scan:"CLOSE 205;STORE 6"
290 OUTPUT @Scan;"OPEN 205"
300 OUTPUT @Scan;"CLOSE 204;STORE 7"
310 OUTPUT @Scan;"OPEN 204"
320 OUTPUT @Scan;"CLOSE 203;STORE 8"
330 !
340 ! -------- チャネル・リスト --------
350 !
360 OUTPUT @Scan;"SLIST 1,2,3,3,4,4,5,5,6,6,7,8.8,0"
370 !     STEPコマンドまたは外部の増分
380 !     入力信号により自動増分される
390 !     ステート用の、スキャン・
400 !     リストをセットアップします。
410 OUTPUT @Scan;"DMODE 1,1,0,1"! 外部増分およびスキャナのチャネル
420 !     閉のためにセットアップします。
430 !
440 ! -------- 測定のセットアップ --------
450 !
460 OUTPUT @Dmm;"PRESET;TARM HOLD"! DMMを通常のPRESET状態に設定し、
461 !     トリガ・アームをホールドします。
462 OUTPUT @Dmm:"MFORMAT DREAL"! データをメモリにIEEE倍精度
463 !      実数型で保存します。
470 OUTPUT @Dmm;"TRIG EXT"! DMMを外部トリガに設定します。
480 OUTPUT @Dmm;"APER 20E-6"! 積分回路アパーチャを20µsに設定します。
490 OUTPUT @Dmm;"TBUFF ON"! トリガ・バッファをオンに設定します。
500 !      does not occur
510 OUTPUT @Dmm;"DISP OFF ! DMM前面パネルのディスプレイをオフにします。
520 OUTPUT @DMM;”DELAY 0”! トリガ・イベントと測定スタート間の時間を
530 !       0秒に設定します。
540 !
550 ! --------- 測定リスト -------------
560 !
570 OUTPUT @Dmm;"SUB 1"! DMMプログラムのスタート
580 OUTPUT @Dmm;"MEM FIFO"!  メモリをFIFO(先入れ先出し)に設定します。
590 OUTPUT @Dmm;"DCV 10" !   DMMをDCVファンクション、最大10Vに設定します。
600 OUTPUT @Dmm;"TARM SGL"! (1) TRIG EXTが満足されれば
610 !  測定シーケンスを開始し、1つのトリガ・
620 !  イベントの発生後に停止します。
630 OUTPUT @Dmm;"TARM SGL"!(2)   シーケンスを繰り返します。
640 OUTPUT @Dmm;"TARM SGL"!(3)   シーケンスを繰り返します。
650 OUTPUT @Dmm;"ACBAND 1000"  !  周波数範囲の下限を1kHzに設定します。
```

```
660 OUTPUT @Dmm;"ACV 10"! DMMの最大入力をAC 10Vに設定します。
670 OUTPUT @Dmm;"TARM SGL"! (3
680 OUTPUT @Dmm;"TARM SGL"! (4)
690 OUTPUT @Dmm:"DCV 10”
700 OUTPUT @Dmm;"TARM SGL"! (4)
710 OUTPUT @Dmm;"TARM SGL"! (5)
720 OUTPUT @Dmm;"ACV 10"
730 OUTPUT @Dmm;"TARM SGL"! (5)
740 OUTPUT @Dmm;"TARM SGL"! (6)
750 OUTPUT @Dmm;"DCV 10"
760 OUTPUT @Dmm;"TARM SGL"! (6)
770 OUTPUT @Dmm;"TARM SGL"! (7)
780 OUTPUT @Dmm;"OHM 3E3"! DMMをΩファンクション、3kΩに設定します。
790 OUTPUT @Dmm;"TARM SGL"! (8)
800 OUTPUT @Dmm;"OCOMP ON"! オフセット補正をオンにします。
810 OUTPUT @Dmm;"TARM SGL"! (8)
820 OUTPUT @Dmm;"SUBEND" ! DMMプログラム・メモリの終り
830 !
840 ! -------- 測定の実行 --------
850 !
860 OUTPUT @Dmm USING "#,K";"CALL 1;"! DMMプログラムを呼び出します。
870 OUTPUT @Scan;"STEP"! 最初のセットアップに移り、DMMをトリガします。
880 !
890 !--測定値をDMMからコンピュータに転送--
900 !
910 FOR I=1 TO 13
920 ENTER @Dmm;A(I)
930 PRINT USING "SD.DDDE";A(I)
940 NEXT I
950 SUBEND
```

付録E　　3458Aを使用した高分解能ディジタイズ

付録E　3458Aを使用した 高分解能ディジタイズ

(Product Note 3458A-2から)

はじめに

システム内で、またはコンピュータを伴うスタンドアローンとして、3458Aは非常に低い歪みと高い分解能で波形をディジタイズできます。最大50kHzの周波数成分を持つ信号の直接サンプリング、および最大12MHzの繰り返し信号に対する分解能16ビット以上のサブサンプリングに十分な測定速度と正確なタイミングを3458Aは提供します。

このプロダクト・ノートでは、以下の内容を説明します。

1. 3458Aを使用した、過渡信号の直接サンプリング。
2. 3458Aを使用した、繰り返し信号のシーケンシャル・サンプリング。
3. スロープおよびレベル・トリガを使用して、必要なところでデータを捕捉。
4. 3458AからHP 9000シリーズ200/300コンピュータへ、測定した信号データを100kサンプル/sで転送。
5. 3458Aのプログラム・メモリを使用して多チャネルの信号を捕捉し、3458Aの読み取り値メモリに保存します。タスクが完了すればHP 9000 シリーズ200/300コンピュータに割り込みをかけて、3458Aからコンピュータにデータを転送し、比較、解析、グラフィック表示を行う。
6. 3458Aオプション005波形解析ライブラリを使用して、ディジタイズした信号の捕捉、解析、表示を行う。
7. 波形のディジタイズと動的性能に関する仕様の解釈。

速度と分解能

- 16ビット @100kサンプル/s
- 18ビット @50kサンプル/s

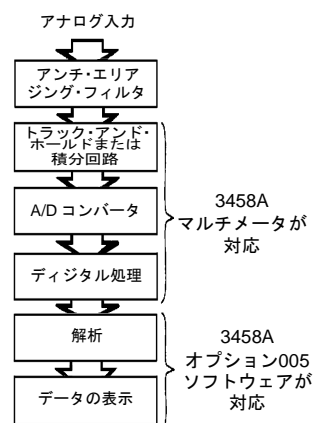
3458Aはオーディオ周波数帯域にわたって、速度と分解能の高い柔軟性を発揮します。DCV測定経路は、175ns未満のトリガ・レイテンシと100ps未満の測定間ジッタにより、オーディオ周波信号をディジタイズします。トラック・アンド・ホールド経路では、3458Aはシーケンシャル・サンプリング(サブサンプリング)を使用し、最大12MHzまでの繰り返し信号を分解能16ビット、50kサンプル/sでディジタイズします。

アナログ信号のディジタイジング

多くのディジタル信号処理システムは、図50のように表すことができます。

どのようなディジタル信号処理システムも、ナイキスト・レートという最小サンプリング・レートがあります。これは、次のようなサンプリング定理によって的確に表現されます。

> アナログ信号をディジタイズするとき、そのサンプリング・レートは、サンプリングする信号のスペクトラムが持つ最大周波数成分(f_0)の2倍以上でなくてはならない。f_0よりも大きな周波数成分は、f_0以下の周波数レンジにエリアスとなって現れ、サンプリング信号の正確な波形と干渉します。例えば、方形波は正弦波の無限和(フーリエ級数)として表現でき、非常に高周波の成分を含んでいます。このような信号を、入力でアンチ・エリアジング・フィルタなしでディジタイズしようとすると多くのエリアスを含むことになり、この信号の波形が意味のない可能性があります。

図50.
一般にディジタル信号処理システムでは、アナログ信号からユーザにとって意味を持つ結果まで、さまざまなファンクションによる解析を必要とします。

エリアジングの防止

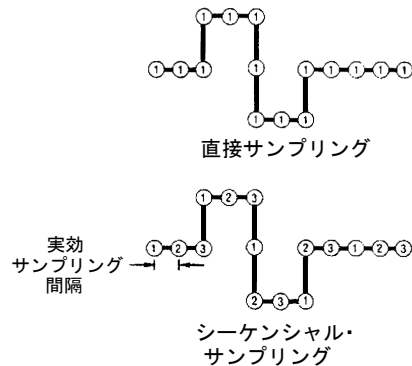
エリアジングにより生じる信号の歪みを回避するには、実効サンプリング間隔がナイキストの定理$1/(2f_0)$を満たさなければなりません。直接サンプリングでは、実効サンプリング間隔は実際の測定間隔です。したがって、トラック・アンド・ホールド経路およびDCV経路(次セクションで説明)において、最大信号周波数はそれぞれ20µs、10µsのサンプリング間隔に対し25kHz、50kHzとなります。もしそれ以上の周波数成分があるときは、帯域幅f_0以下のローパス・フィルタを信号経路に挿入する必要があります。

シーケンシャル・サンプリングでは、実効サンプリング間隔は再構築された波形上のサンプル間の間隔です(図51)。実効サンプリング間隔の設定が35ns以下のときは、トラック・アンド・ホールド経路の帯域幅12MHzによって、エリアジングによる歪みがほとんど除去されます。実効サンプリング間隔が35ns以上で、12MHz以上の周波数成分が存在する場合は外部フィルタが必要です。

図51. 直接サンプリングは、信号の1回の通過で波形を捕捉します。シーケンシャル・サンプリングは繰り返し信号を必要とし、信号の何回かの通過によって周期が再構築されます。図中の番号は、1周期の入力によって捕捉されたサンプルを示しています。

直接サンプリング

実効サンプリング間隔

シーケンシャル・サンプリング

2つの測定経路の比較

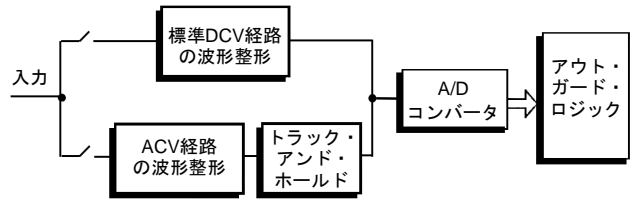
3458Aは標準DCV経路とトラック・アンド・ホールド経路の、2種類の入力測定経路を備えています(図52)。トラック・アンド・ホールド経路は、サブサンプリングと直接サンプリングに使用されます。DCV経路は、直接サンプリングのみです。標準DCV経路をサブサンプリングのために使用することも可能ですが、データ捕捉のためにアルゴリズムをプログラムする必要があります。

DCV経路を直接サンプリングに使用

標準DCV経路は、コマンド"PRESET DIG"をプログラムした場合に選択されます。このコマンドにより、入力信号を直接ディジタイズするためのデフォルトのパラメータが設定されます(256サンプル、50kサンプル/s、フル・スケール、10Vピーク)。トリガ回路では、0Vレベル、立ち上がりスロープでトリガ、AC結合が設定されます。これにより、これらのデフォルト設定によって200Hz〜25kHzの少なくとも1波形が捕捉できます。

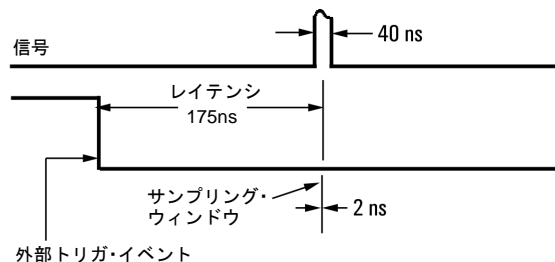
また標準DCV経路では、18ビット(5 l/2桁)@6kサンプル/sから16ビット(4 l/2桁)@100kサンプル/sの間で、速度と分解能をトレードオフできます。10Vレンジにおけるノイズ・フロアは、それぞれ0.005%および0.05%です。

DCV経路で分解能を高くすると、アパーチャ時間も増加します。したがって、低ノイズおよび高分解能にすると、サンプリング・アパーチャが拡大し、情報の損失をもたらします。パルスのピーク値を捕捉するには、アパーチャはパルス幅よりも広くてはいけません。実用的には、サンプリング・アパーチャの幅に近いパルスでは、トリガの不確かさによりピーク振幅の捕捉がほとんど不可能になります。解決法としては、入力増幅器の帯域幅を、サンプリング・アパーチャではなく、分解能の制限要因となるポイントまでアパーチャを狭めます。

図52. 3458Aマルチメータは、標準のDCV経路とトラック・アンド・ホールド経路の、2種類のディジタイズ経路を備えています。

トラック・アンド・ホールド経路を直接またはシーケンシャル・サンプリングに使用

トラック・アンド・ホールド経路は、狭いパルスを捕捉するための機能です。この経路の帯域幅は12MHzで、アパーチャは2nsに固定されています。2nsのトリガ・ジッタで、測定の劣化なしに、少しのサーチ時間で、40nsという短いパルスのピーク振幅も捕捉できます(図53)。10nsに満たない立ち上がり時間は、ディジタイズ測定にオーバシュートを発生させます。したがって、そのような周波数成分を持つ信号を3458Aの入力に加える場合は、フィルタリングによってその信号を帯域制限する必要があります。トラック・アンド・ホールド経路による直接ディジタイズでは、最大12MHzの周波数成分を持つ信号を捕捉できます。またこの経路により、最大12MHzの繰り返し信号をサブサンプリングできます。

3458Aのトラック・アンド・ホールド経路を使用した、直接およびサブサンプリング(シーケンシャル)ディジタイズのためのプログラミングは簡単です。一つのコマンドしか必要ありません。例えばDSACはAC結合の直接サンプリングを、SSACはAC結合のシーケンシャル・サンプリングを行います。これらのコマンドは自動的にデフォルトのパラメータを使用しますが、そのパラメータは変更可能です。

図53. 狭いパルスを捕捉するには、12MHzトラック・アンド・ホールド経路の使用が必要です。サンプル捕捉とトリガ・イベント間の最小時間は175nsです。

データの捕捉

測定サイクル開始のために3458Aをトリガするには、入力信号のレベルおよびスロープ、電源ラインの0Vレベル交差、GPIBからのGET(Group Execute Trigger)コマンド、外部TTL信号、内部生成のトリガ信号(バースト測定のときペース調整が可能)、またはコンピュータによる測定要求を使用できます。

3458Aでは必要な信号を捕捉するために、3レベルのトリガ、および波形のレベルとスロープを含む8つまでの条件を使用できます。トリガ・レベルの階層はトリガ・アーミン

グ(TARM)、トリガ(TRIG)、1トリガ当たりの読み取り回数(NRDGS)です。直接サンプリングとサブサンプリングのために、NRDGSに関係したSWEEP、およびサブサンプリングのためのトリガ・ソース(レベルまたは外部)を選択するSSRCの、ディジタイズ専用の2つのコマンドを使用できます。また図54に示すような、測定の実行前に満足させる必要のあるさまざまなイベントや条件を設定できます。トリガの3つのレベルに対するデフォルト条件は、AUTOです。このとき、3458Aはそのセットアップで可能な最速のトリガを生成します。

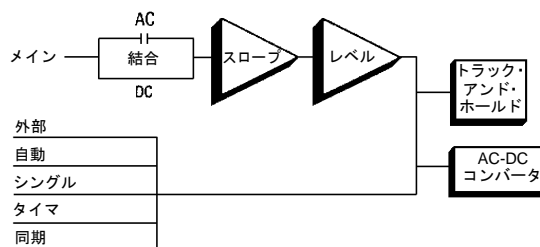
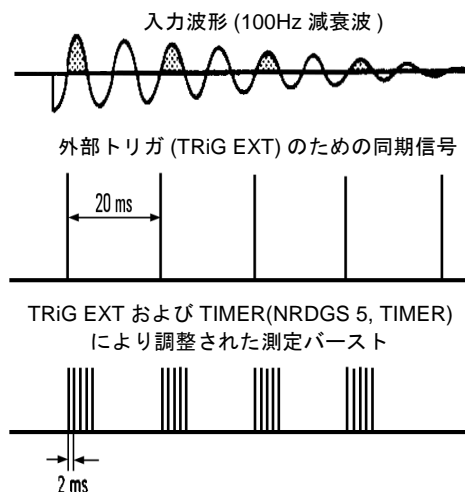
図54. これらのトリガ・イベントの選択により、さまざまなアプリケーションに対応できます。

TARMは、満足させる必要のある最初の条件です。これは、トリガ信号を受信する前にトリガ回路をアームします。例えば目的の信号以外に同期信号が利用できる場合、TARM EXTはアームするためにそのトリガ・イベントを待ちます。TARMはまた、特定の測定サイクルの繰り返し回数を指定して、複数の測定シーケンスの制御に使用できます、例えばTARM SGL,4は、トリガ・アーミングを4回繰り返した後に停止します(図55)。

図55. 標準トリガ・コマンドによるディジタイズ。トリガ・アーミング・コマンドTARM SGL,4により、測定サイクルを4回だけ行なって、入力波形内の影で示した領域の比の決定に要するデータ量を低減できます。

```
10 OUTPUT 722; “TARM HOLD”              ! 3458を測定ホールド状態に
                                          設定します。
20 OUTPUT 722,”TRIG EXT”                ! トリガを起すトリガ・イベントを
                                          設定します。
30 OUTPUT 722, “NRDGS 5, TIMER”         ! 各トリガに5回の読み取りの
                                          バーストを設定します。
40 OUTPUT 722; “TIMER 2E-3”             ! 読み取り間隔をTIMER設定
                                          (2E-3または2ms)にします。
          …                             ! 外部トリガの発生により、
200 OUTPUT 722; “TARM SLG, 4”             4回の測定バーストを行います。
```

TRIGは次に満足させる必要のある条件です。TARMおよびTRIGイベント条件の両方を満足した後にだけ、NRDGSによってバースト測定を実行できます(図56)。コマンド

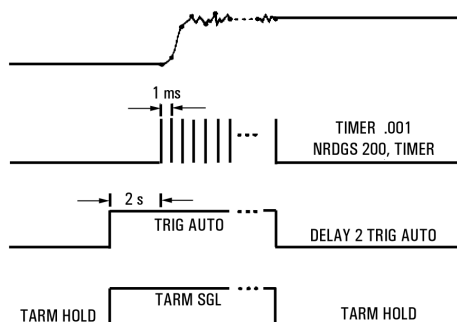
```
NRDGS [# 読み取り回数] [,イベント]
```

は読み取り回数、読み取りのためのトリガ条件、およびトリガ・イベントの前後にメモリにセーブする読み取り数を設定します。

SWEEPおよびSSRCは、ディジタイズ・タスクを容易にするための特別なコマンドです。コマンド

```
SWEEP [測定間の実効間隔] [, 測定数]
```

はNRDGSパラメータをTIMERと組み合わせます。SSRCは、サブサンプリングのための同期信号源(外部またはレベル)を選択します。SWEEPおよびSSRCコマンドは、SSAC(サブサンプリング、AC結合)、SSDC(サブサンプリング、DC結合)用に使用され、NRDGSおよびTRIGは無視されます。DSAC(直接サンプリング、AC結合)およびDSDC(直接サンプリング、DC結合)に対しては、すべてのトリガ・コマンドが有効ですが、両方を同じ測定で使うことは推奨されません。

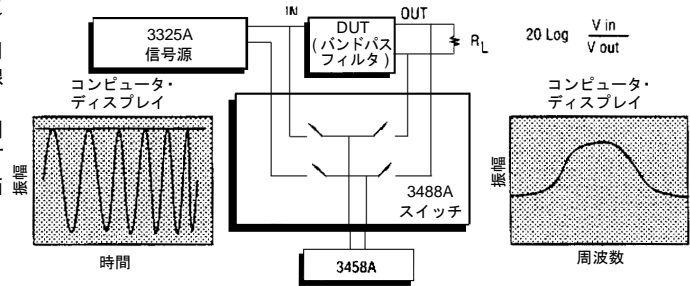
図56. 図で示すように、トリガ・アームおよびトリガ・イベント条件が満足されれば、バースト測定による波形のディジタイズが可能です。

SSRCコマンドによって、内部レベルによるトリガ、または外部トリガとの同期が可能です。サブサンプリング・モードで、SSRC EXTはSWEEPコマンドで指定する測定の実行に必要な外部トリガの個数を計算します。

例えば100nsの時間分解能で4096回の読み取りを行なって波形を捕捉するときは

```
[SWEEP 100E-9,4096]
```

3458Aは読み取り数に時間間隔を掛けて、サンプリング間の最小時間で割ります。

遅延を外部トリガ同期とともに使用して、測定に時間窓を適用して、解析したい波形を部分的に調べることができます。例えば、3458Aを3325A信号源とともにゲイン／フェーズ・メータとして使用し、周波数レンジ0.5～5MHzで帯域通過フィルタの伝達関数の測定を考えます(図57参照)。最大周波数は5MHzのため、1周期2サンプリングでは、全帯域に対して、サンプリング間の最小時間は100nsです。この解析のためには、2つの方法が考えられます。(1)周波数スペクトラム全体を、100nsの間隔で掃引します。または、(2)周波数スペクトラムをいくつかの帯域に分けて、それぞれの帯域に対して1/(2f_0)で掃引します。最初の方法ではデータ捕捉の時間が最も短くなり、2つめの方法では高速なコンピュータの必要性が最も小さくなります。

図57. 掃引周波数発生器とともに、3458Aをフェーズ／ゲイン・メータとして使用して、振幅のみのボード線図プロットを表示します。位相同期トリガによって測定のタイミングを得て、DUTを周波数に対して特性評価できます。

高速データ転送

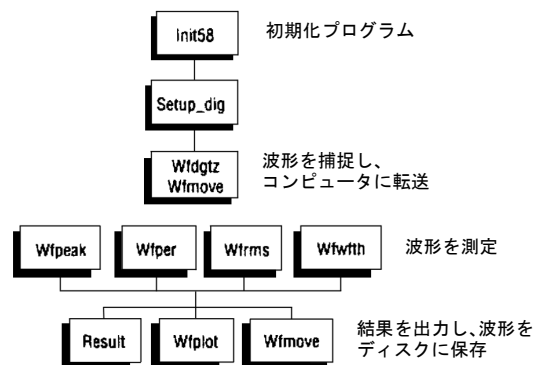
3458Aはその最大の読み取り速度で、読み取り値をDMA(ダイレクト・メモリ・アクセス)カードを備えた9000 シリーズ200/300コンピュータに転送できます(コンピュータをその速度での捕捉を可能なようにセットアップしている場合)。測定値は3458Aの内部メモリから、または3458Aが測定を行うのと同時に、取り出すことができます。この場合、3458Aが高速測定に専念できること、コンピュータが適切なバッファを持つことの、2条件が必要になります。

PRESET DIGは、まさにそのために3458Aが必要とするコマンドです。このコマンドは、DMMを最大限の速度にセットアップします。

直接ディジタイズを選択している限り、コンピュータ画面での波形の再構成に問題はありません。転送する前にデータの保存場所としてメモリを使用している場合、3458Aはデータの順序を考慮して保存します。

波形解析ライブラリ

3458Aオプション005の波形解析ライブラリ(03458-80005)を使えば、3458Aの制御コマンドを一つも使わずに波形捕捉が可能なだけでなく、コンピュータや測定器の知識が最少限でもデータを解析／表示できます。マスター・プログラムの作成では、測定セットアップ、波形捕捉、解析、表示の簡単なシーケンスを考えるだけで済みます。このマスター・プログラムが、BASIC言語とコンパイルされたサブプログラムを呼び出します(図58参照)。

図58. ライブラリのサブプログラム(完全なプログラム・リストである必要はありません)を使用した自動測定プログラムの典型的な構築図。

波形解析ライブラリは周波数、立ち上がり時間、パルス幅、オーバシュートなどの時間ドメイン解析に加えて、高速フーリエ変換(FFT)および逆フーリエ変換(IFT)、ハニング・フィルタによる周波数ドメイン解析も提供します。また波形解析ライブラリは、3458AとHP 9000 シリーズ200/300コンピュータにより最速50,000サンプル/sの測定を行い、最大5フレーム/sのリフレッシュ・レートでコンピュータ画面に表示する「高速スコープ」プログラムも提供します。これらの組み合わせは、実質的に非常に高分解能の12MHzシングル・チャネル・オシロスコープとして使用できます。

また波形解析ライブラリでは、測定のリミット付きの捕捉済みの波形を、入力信号と比較できます。

さらに波形解析ライブラリでは、次のようなユーティリティ機能も提供します。Format－出力画面を工学単位でフォーマットします。Intrpo－サンプル・ポイント間のリニア補間を行います。Sinc－ナイキスト・リミット近くで捕捉された信号に対し、サンプル・ポイント間のsinc関数補間を行います。Warn58－コンピュータのモニタまたはプリンタに、エラー／警告メッセージを出力します。

ここで、波形解析ライブラリを使用して、AM変調信号を捕捉し、搬送波、変調周波数、および変調の深さを抽出する例を示します。

最初に、ライブラリのサブプログラムを呼び出す、メイン・プログラムを作成する必要があります。メイン・プログラムは、測定タスクが必要とする順序でサブプログラムを制御・起動する、プログラム・コードのブロックです。メイン・プログラムは測定タスクのニーズによって、長くも短くもなります。その例の一部を、下に示します。このプログラムは3458Aを使用して波形を捕捉し、それをコンピュータに転送し、コンピュータのディスプレイでプロットします。ここでは、次の4つのサブプログラムを使用します。Setup_dig－DMMのセットアップ・サブプログラムで、ディジタイズの方法(DCV、DSAC、DSDC、SSAC、SSDC)、サンプリング間の時間間隔、サンプリング数を設定します(FFTまたはIFTルーチンを使用する場合、サンプリング数は2の累乗の必要があります)。Wfdgtz－波形捕捉サブプログラム。Wfmove－転送サブプログラム。Wfplot－プロット・サブプログラム。

```
1280 CALL Setup_dig(1,1.e-5,1000)
1270 CALL Wfdgtz(1)
1280 CALL Wfmove("1","98",Scal(*),Wavf(*),Clip)
1290 CALL Wfplot(Scal(*),"Wave form 1",Wavf(*),1,1)
```

サブプログラムは、どのプログラム言語でも最も強力な要素の1つです。サブプログラムはそれぞれ、メイン・プログラムとは別のコンテキスト、ステートを持ちます。つまりそれは、独自の変数セット、行ラベルを持つことを意味しています。

スターター・メイン・プログラム

ライブラリのサブプログラムを使用するすべてのプログラムは、メイン・プログラムが必要です。本章で説明しているデータ配列の多くは、各メイン・プログラムで宣言する必要があります。また多くのライブラリ・サブプログラムで使用されるCOM文も、メイン・プログラムの多くで必要です。波形解析ライブラリには、メイン・プログラム全体の開始に使用できる、次のようなスターター・メイン・プログラムが含まれています。

```
 10  ! Main
 20  ! Core main program programming aid
 30  !** COMMON
 40  COM/Hp3458/@Recorder,Xist_plotter,Prt,Bus,Xist
 50  !** Real Arrays
 60  REAL Scal(0:4),Yamp(0:7)
 70  !** STRINGS
 80  DIM Source$[50],Destin$[50],Titles$[30]
 90  !** INTEGER ARRAYS
100  INTEGERWavf(1:16384),Redg(0:30),Fedg(0:30),Bandwf(0:163)
110  DISP                                ! ディスプレイ・ラインをクリア
120  OUTPUT I USING "@"                  ! CRTをクリア
130  !
140  CALL Init58                         ! バスを起動
150  !
160  GINIT                               ! グラフィックを初期化
170  !
180                                      ! メイン・プログラムをここから
250                                      ! ここに挿入
260  END
```

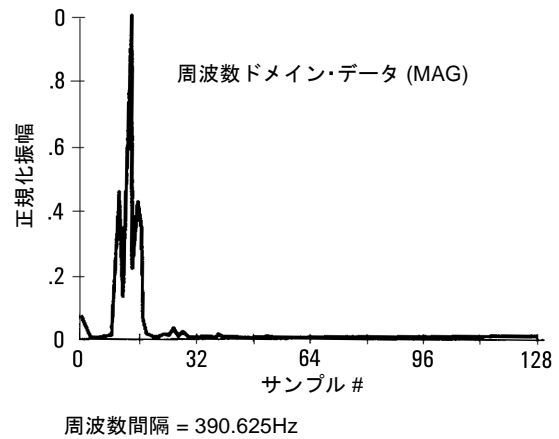
もとのプログラムに戻ると、AM変調信号の解析に必要なサブプログラムは、以下です。

Setup_dig、Wfdgtz、Wfmove、Fft、Fft_plot

つまり、以下をメイン・プログラムに挿入します。

```
190 CALL Setup_dig(2,20E-6,512)
200 CALL Wfdgtz (1)
210 CALL Wfmove("1","98",Scal(*),Wavf(*),Clip)
220 CALL Fft (512,1,Hanning,Wavf(*),Real_dat(*),Imag_dat(*),Magn_dat(*))
240 CALL Fft_plot(Magn_data(*),Smpl_intvl,Dyn_range,F_start,F_stop,Title$)
250 END
```

このプログラムの結果を図59に示します。

図59. 波形解析ライブラリを使用した測定結果の例

測定誤差

3458Aの柔軟性によって、ディジタイズ・プロセスで生じる測定誤差の多くを防止・補正することができます。ディジタイズに関連する誤差は振幅誤差と時間誤差に分けることができ、それらが合わさって測定全体の誤差となります。また動的な信号では、時間誤差が振幅誤差となって現れます。幸いにも、ほとんどの時間依存測定は差動であり、絶対的な時間誤差は校正で測定から除かれます。3458Aのブロック図をよく見ると、図60で示すような測定誤差の発生源が分かります。

一般的に言って、ディジタイズ測定に潜む誤差は、振幅上および時間軸上ともに明白です。

振幅の誤差要因には、次のものがあります。

1. 量子化誤差
2. 欠落コード
3. 非線形性
4. ノイズ
5. 帯域幅
6. 振幅確度

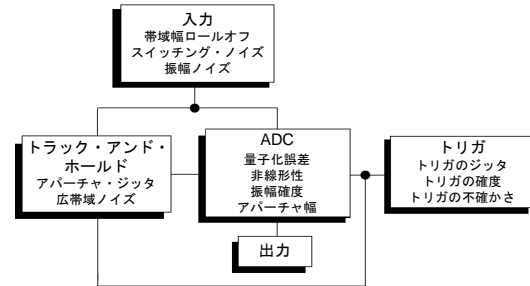
時間軸上の誤差要因には、次のものがあります。

1. タイムベース基準のジッタ
2. トリガの不確かさ
3. トリガの確度

4. トリガのレイテンシ

5. アパーチャ幅

6. アパーチャ・ジッタ

図60. これらのディジタイズ誤差の発生源は、どのような測定でも考慮する必要があります。

振幅誤差

3458Aの入力信号整形セクションには、A/Dコンバータ(ADC)またはトラック・アンド・ホールドのための信号の整形とルーティングに関係した、スイッチ(リレー)、減衰器、増幅器があります。オートゼロは入力オフセット誤差を除去しますが、残留誤差は伝搬されます。このセクションは、3458Aの低周波数セクションです。したがってそのレンジによっては、信号はADCに送る前にローパス・フィルタ(入力増幅器)を通ります。

量子化誤差は、連続した(アナログ)信号を有限のディジタル・ビットへ完全量子化することに関係した、本質的で低減不可能な誤差です。したがってADCの分解能が、入力波形をいかに細かく測定できるかを直接左右します。大きなオフセットがあっても信号の詳細な解析を可能にするウィンドウ増幅器によって、いくらかの制限を除けるかもしれません。しかし増幅器の導入は、高分解能ADCに不必要な測定誤差を加えることになります。

欠落コードは、高速においてのみ現れます。欠落コードの最も一般的な原因は誘電吸収(DA)で、これは導線周囲の絶縁材のダイポール分極です。設計段階の対応でこれは防げますが、DAは測定値に前の測定のメモリ効果を生じさせることがあります。ADCに十分なセトリング時間を与えれば、この問題を量子化レベル以下に抑えることが可能です。

欠落コードと量子化誤差が組み合わさると、ADCの非直線性が生じます。これは微分非直線性と積分非直線性の、2つの形で現れます。微分非直線性は、連続した量子化レベルの間で生じる最大のステップです。積分非直線性は、最小2乗法フィッティングからの、直線性カーブの最大偏差です。一般に微分非直線性は、低レベルの信号が微分非直線性誤差を持つADC伝達関数の領域に当ったときに、大きな測定誤差を引き起すことがあります。ADCの積分非直線性は一般に、フル・スケール信号のディジタイズでデメリットになります。

ADCの伝達関数が、スルーレート(dV/dt)に非常に依存することを理解してください。静的DC入力レベルに対する伝達関数は、理想的になります。これに対し動的な状況下での伝達関数は、多くの誤差を示すことがあります(図61)。

帯域幅増加に付随するノイズは、どのような測定でも不可避です。ランダム測定ノイズの影響は、測定値のアベレージングによって低減することができます。ジョンソン・ノイズや他の回路関連のノイズ、および入力信号上のノイズにより生じるこのノイズの除去は、常に測定時間を増加させます。ディジタイズ装置の品質尺度としての、分解能の「有効ビット」は、ノイズをADCの直線性と組み合わせてディジタイザの実効分解能を示します。

有効ビット = N - log_2 (実効誤差(実際) / 実効誤差(理想))

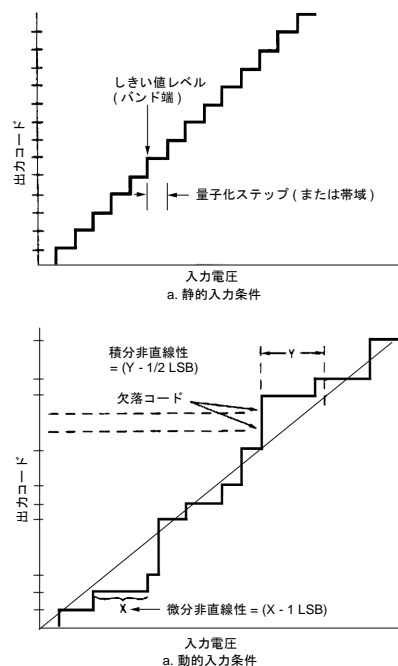
図61. 静的DC入力レベルに対し、A/Dコンバータは、12aに示すような理想的な伝達関数を示します。しかし動的な入力では、12bで示すような誤差が現れます。

実効誤差(実際)は、最適フィッティングの完全な正弦波に対して測定した誤差です。実効誤差(理想)は、完全なNビットADCの理論上の誤差です。低分解能の測定器では、有効ビットが真の測定品質です。これに対し高分解能の測定器では、測定に関連したノイズはADCの実際の性能を低下させます。しかしサンプリング数を多くしたり、サンプルをアベレージングすれば、実際の量子化および非直線性ノイズがサンプリングしたデータのフーリエ変換において明白になるポイントまで、ノイズを低減することができます。この効果を、図62で示しています。入力信号の第三高調波が、実際の積分非直線性です。10サンプルのアベレージングによってそのレベルは除去されませんが、ノイズ・レベルは10dB低下しています。

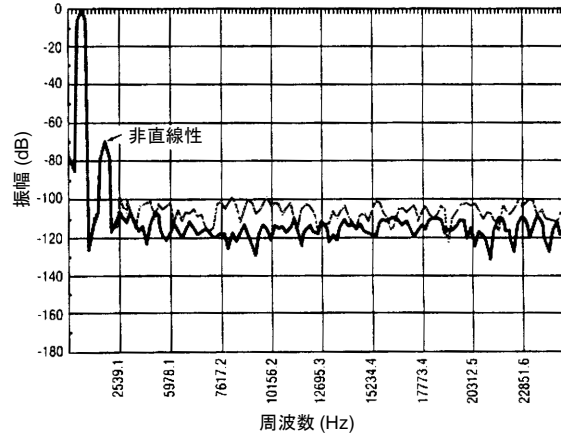
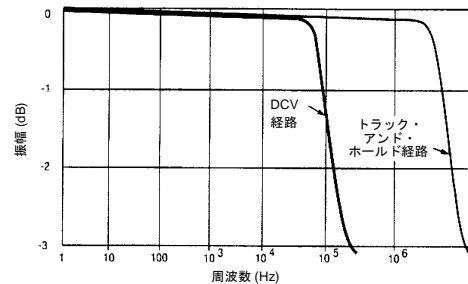
図62. 非直線性誤差を示すA/Dコンバータが、アベレージングによって除去されないスプリアス応答を発生させます。3458Aは16ビット@100,000読み取り/sまでリニアです。

3458Aは、2つの入力経路を提供します。2つの経路の違いは、直接ADC経路(DCV)は最大帯域幅160kHz、サンプリング・レート100,000サンプル/sを提供し、トラック・アンド・ホールド経路は最大帯域幅12MHz、サンプリング・レート50,000サンプル/sを提供することです。双方の経路とも単極ロールオフを示し、また帯域幅ポイントにおいて公称3dB低下(電力半値)します。したがってエリアジングと振幅ロールオフの、2種類の誤差の可能性があります。トラック・アンド・ホールド経路では、実効サンプリング・レートを100Mサンプル/sに増加することによりエリアジングを除去できます。また、トラック・アンド・ホールド回路は必要な帯域にわたって振幅ロールオフを評価でき、ロールオフの補正が可能です。DCV経路の場合、エリアジングに対する実際的な解決法は、ローパス・アナログ・フィルタを加えることだけです(図63)。

図63. 3458Aマルチメータの2つの測定経路に対する振幅ロールオフ

最後に、ディジタイザに関してあまり論じられませんが、測定自体の確度は3458Aの基準確度に関係します。静的および動的測定について、絶対確度は実際には3458Aの分解能を上回っています。また長期ドリフトについては、絶対誤差は7ppm/年未満となっています。

トリガおよびタイムベースの誤差

タイムベースの精密温度補償型クォーツ・クリスタルは、入力信号の測定に影響する、それ自体のドリフトおよびジッタを持ちます。しかしこれらは、50ps未満と非常に小さな値です。したがってクロック確度とジッタは、3458Aの測定帯域幅において、測定に実際上の影響を与えません。タイムベースのジッタ誤差は、累積しません。したがって各サンプリング・ポイントはそれ自体のジッタ誤差のみを持ち、前のサンプリング・ポイントと合わせた誤差はありません。全時間軸誤差の影響を、図64で示しています。

トリガ誤差は、タイムベース誤差およびジッタを数桁上回る大きさです。2つの影響でこれが生じます。3458Aは遅延ラインを持たないため、トリガと測定開始間の時間遅延としてのトリガ・レイテンシがあり、これはファームウェア、クロック、タイミング回路によって決まっています。その値は、外部トリガに対して175ns未満に仕様化されています。またトリガ確度は、トリガ信号上のノイズ、および時間補間器の測定間での変動の影響を受けることがあります。これも、非常にノイズの多いケースを除いて50psのオーダとなっています。ノイズの多いケースでは、3458Aのトリガ・フィルタの使用を推奨します。このトリガ・フィルタはトリガ回路の帯域幅を、70kHz(公称)に減らします。

図64. タイムベース・ジッタの影響。3458Aマルチメータでは、ジッタは50ps RMSです。このジッタは再現性を持つため、特性評価による補正が可能です。

索引

数字

A

B

C

D

E

F

G

H

I

L

M

N

U

W

あ

か

さ

た

な

は

ま

や

ら

わ

納入後の保証について

- ハードウェア製品に対しては部品及び製造上の不具合について保証します。又、当社製品仕様に適合していることを保証します。
  ソフトウェアに対しては、媒体の不具合（ソフトウェアを当社指定のデバイス上適切にインストールし使用しているにもかかわらず、プログラミング・インストラクションを実行しない原因がソフトウェアを記録している媒体に因る場合）について保証します。又、当社が財産権を有するソフトウェア（特注品を除く）が当社製品仕様に適合していることを保証します。
  保証期間中にこれらの不具合、当社製品仕様への不適合がある旨連絡を受けた場合は、当社の判断で修理又は交換を行います。

- 保証による修理は、当社営業日の午前８時45分から午後５時30分の時間帯でお受けします。なお、保証期間中でも当社所定の出張修理地域外での出張修理は、技術者派遣費が有償となります。

- 当社の保証は、製品の動作が中断されないことや、エラーが皆無であることを保証するものではありません。保証期間中、当社が不具合を認めた製品を相当期間内に修理又は交換できない場合お客様は当該製品を返却して購入金額の返金を請求できます。

- 保証期間は、製品毎に定められています。保証は、当社が据付調整を行う製品については、据付調整完了日より開始します。但し、お客様の都合で据付調整を納入後 31 日以降に行う場合は 31 日目より保証が開始します。
  又、当社が据付調整を行わない製品については、納入日より保証が開始します。

- 当社の保証は、以下に起因する不具合に対しては適用されません。

  (1) 不適当又は不完全な保守、校正によるとき
  (2) 当社以外のソフトウェア、インターフェース、サプライ品によるとき
  (3) 当社が認めていない改造によるとき
  (4) 当社製品仕様に定めていない方法での使用、作動によるとき
  (5) お客様による輸送中の過失、事故、滅失、損傷等によるとき
  (6) お客様の据付場所の不備や不適正な保全によるとき
  (7) 当社が認めていない保守又は修理によるとき
  (8) 火災、風水害、地震、落雷等の天災によるとき

- 当社はここに定める以外の保証は行いません。又、製品の特定用途での市場商品価値や適合性に関する保証は致しかねます。

- 製品の保守修理用部品供給期間は、製品の廃止後最低５年です。

原　典

本書は"Agilent Technologies 3458A Multimeter User's Guide" (Part No. 03458-90014) (Printed in U.S.A, December 2000)を翻訳したものです。

詳細は上記の最新マニュアルを参照して下さい。

ご　注　意

- 本書に記載した内容は、予告なしに変更することがあります。
- 当社は、お客様の誤った操作に起因する損害については、責任を負いかねますのでご了承ください。
- 当社では、本書に関して特殊目的に対する適合性、市場性などについては、一切の保証をいたしかねます。
- また、備品、パフォーマンス等に関連した損傷についても保証いたしかねます。
- 当社提供外のソフトウェアの使用や信頼性についての責任は負いかねます。
- 本書の内容の一部または全部を、無断でコピーしたり、他のプログラム言語に翻訳することは法律で禁止されています。
- 本製品パッケージとして提供した本マニュアル、フレキシブル・ディスクまたはテープ・カートリッジは本製品用だけにお使いください。プログラムをコピーをする場合はバックアップ用だけにしてください。プログラムをそのままの形で、あるいは変更を加えて第三者に販売することは固く禁じられています。